AD-A080 371   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: --ETC(U)
DEC 79 J R MCCLENDON
UNCLASSIFIED AFIT/GE/EE/79-22 NL

1 OF 3
AD
A080371

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GE/EE/79-22

6 MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: THEORY, APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION.

9 Masters THESIS

Robert

14 AFIT/GE/EE/79-22

10 James / McClendon
Lieutenant USAF

11 Dec 79

12 214

Approved for public release; distribution unlimited

012225

AFIT/GE/EE/79-22

MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: THEORY APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

By

James R. McClendon, B.S.
Lieutenant USAF

Graduate Guidance and Control
December 1979

Approved for public release; distribution unlimited.

PREFACE

Many model order reduction techniques exist. However, most of these techniques, though based on sound theory, require many man hours of analysis and prior system insight to perform. This thesis implements a model order reduction technique developed by Dr Bruce Moore of the University of Toronto and applies the resulting software package to find a reduced order model of a B-52E flutter control problem which is currently of concern to the Air Force Flight Dynamics Laboratory. This particular technique utilizes a well defined, highly efficient algorithm resulting in impressive reduced order models. Although other techniques may be found providing a reduced model of similar order, this technique has a distinct advantage in that it is highly systematic and may be totally computer automated through the use of "high quality" software routines of IMSL, Eispack or Linpack (Ref 13).

The model order reduction technique currently in use by the Air Force Flight Dynamics Laboratory, Flight Control Division under Dr Bob Schwanz, requires human decisions to determine which modes of a system to retain. The technique to be investigated herein requires no such human decision.

An interactive computer program employing Dr Moore's algorithm as well as offering other related options is presented in the hope the user may obtain reduced order models representing his full order system and investigate the reduced order model's robustness properties in an efficient manner.

I wish to thank Lt Stanley J. Larimer for his help in interfacing with TOTAL to obtain the time and frequency domain responses that were crucial for evaluation of this model order reduction technique. I would like to thank Capt Jerry Stinson for his help and assistance in learning how to use the ASD Computer System.

My sincere gratitude is extended to Major J. Gary Reid, without whose extensive knowledge and genuine zeal for this project would have made it an impossibility. I would like to also thank Dr Bob Schwanz of the Air Force Flight Dynamics Laboratory for his help and sponsorship of this project.

I wish to thank my wife, Kathy, and my son, Jake, for their understanding and encouragement when I really needed it. Finally, a note of appreciation is necessary for my typist, Eve Vaught, for her outstanding work in typing this thesis.

James R. McClendon

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

LIST OF FIGURES

LIST OF FIGURES
(continued)

LIST OF TABLES

LIST OF SYMBOLS

Symbol	Meaning
$\sigma(i)$	The i th singular value.
A^T	The matrix A transposed.
A^H	The conjugate transpose of A.
$K(A)$	The condition number with respect to inversion of A.
$\|\cdot\|$	Euclidean norm for a vector Subordinate spectral norm for a matrix.
A^+	Generalized inverse of A.
A^{-1}	The inverse of A (square).
$S^\perp$, $S \epsilon R^n$	Orthogonal complement in R^n.

ABSTRACT

This report investigates a model order reduction technique developed by Dr Bruce Moore of the University of Toronto, as applied to the B-52E flutter control problem currently under study by the Air Force Flight Dynamics Laboratory. The algorithm, which is based upon singular value analysis, is applied to the full twenty-fourth order model yielding an internally balanced representation which is balanced with respect to controllability and observability properties. The system is reduced in order, and comparisons are made between the Moore algorithm model and that obtained via a method used by the Air Force Flight Dynamics Laboratory.

In addition to this investigation, an interactive program is presented which contains the model order reduction algorithm. Other capabilities include: estimation of the condition number with respect to inversion, singular value and condition number plotting vs. sample time for discrete time controllability, observability, and Hankel matrices, frequency response generation, and various special coordinate system transformations.

I. INTRODUCTION

High speed digital computers allow us to investigate systems faster than ever before possible. However, these computers have memo:y and central processor utilization time restrictions. Many systems such as chemical processes and mechanical structures have large dimension, complex system models. Some systems, such as temperature control systems, are described by partial differential equations. Such systems are infinite dimensional and are called distributed parameter systems. Another example of a distributed parameter system is the flexible wing on an airplane. These partial differential equations must be transformed to ordinary differential equations to yield a finite system model. Often these system models exceed the computer's memory restrictions and a lower order model must be found. For a typical distributed parameter system, system orders in the range of 100 or more are common and are considered large dimension systems.

Two applications for reduced order models are simulation and on-line control. In simulation, the desire is possibly to test equipment or algorithms with the system model. To be feasible in terms of cost and turnaround time, it is highly desirable to utilize a reduced order model which replicates the actual system accurately while being of sufficiently small dimension so as to allow cost effective simulation.

The other application is on-line control. This occurs when a controller is trying to control an actual system on-line in real-time. The model used for control calculations must be small enough so as not to exceed the computation time required for effective control. Often in on-line control, a Kalman filter is required to estimate the states of the system. A reduced order model can be highly beneficial to a Kalman filter design also.

The hardest task in designing a Kalman filter is obtaining an accurate representation of the system (i.e., the model). Unfortunately, the system order is often too great for efficient Kalman filter implementation. For an n-dimensional system, where n is large, the number of multiplies that must be accomplished by a Kalman filter is proportional to n^3 (Ref 22). The Kalman filter gain requires the solving ot n(n+l)/2 simultaneous equations. Clearly, if the model's order could be reduced while still yielding "good" results, computational burden could be decreased substantially. As an example, if the original system is twenty-fourth order (24 state variables), then 300 simultaneous equations must be solved to calculate the Kalman filter gain. If the model could be reduced to tenth order, only 55 simultaneous equations need be solved.

Other applications such as controller design and actual on-line controller implementation require models of sufficiently low order.

Therefore, the need exists to efficiently and accurately reduce the order of a system model without a prohibitive loss of accuracy.

Background and Problem Statement

Many model order reduction techniques exist. Rogers and Sworder (1971) chose model parameters to optimize a certain cost functional. Hutton and Friedland (Ref 17) suggest the use of Routh approximants. A Routh approximant is defined by making the Routh table coefficients for the approximant match those of the original system to a given order. Chien and Shieh (1968) propose the use of Pade' approximants. A Pade' approximant is defined by choosing the coefficients such that the Taylor sines expansions of the approximant and the original system agree in as many terms as possible for a predetermined system order lower than that of the original system. Many "dominant mode retention" procedures including that of Chidambara (1967) and Marshall (1966) exist. Norm minimization techniques such as the one proposed by El-Attar and Vidyasagar (Ref 14) can be found in the literature. The list is long. The main point, however, is that many techniques exist.

Some of these techniques result in unstable reduced order models when given a stable full- order model. Some are not easily implemented on a computer, and thus many man hours must be spent in deriving the reduced order model. Some techniques work with state space models, others with transfer functions, and still others deal with the full-order differential equations of the full-order system.

The problem to be investigated in this paper is to implement and analyze a model order reduction algorithm developed by Dr Bruce Moore of the University of Toronto (Ref 24, 25). This algorithm minimizes many of the aforementioned problems associated with model order reduction techniques. The Moore algorithm actually implements working subspaces for the Kalman minimal realization decomposition (Ref 19).

The Moore algorithm actually contains four paths that may be followed, all of which yield reduced order models. These paths include:

1) A reduced order model that is "optimal" in response to an impulse input (optimal in the sense that this particular model more closely resembles the original system if both are subjected to an impulse input).

2) A reduced order model that is "optimal" in response to a step input.

3) A reduced order model which weights the steady state portion of the response to a given input more heavily than the transient portion.

4) A reduced order model which weights the transient portion of the response to a given input more heavily than the steady state portion.

Statement of Purpose

The purpose of this thesis is twofold. The first objective is to develop software, both interactive and batch, which implements the Moore algorithm and investigates its properties in both the frequency and time domains.

The second objective is to take this software and apply it to a practical Air Force Flight Dynamics Laboratory (AFFDL) problem, specifically that of finding an accurate reduced order model for the B-52E flutter control problem.

Scope and Approach

The objectives are accomplished in the following manner. First, batch software is developed to obtain the B-52E flutter control problem's state space model. Next, an interactive program called MIMO is developed to investigate the application of the Moore algorithm to the full order system model. The actual application of the Moore algorithm to the full order system will be accomplished by a batch program because the full order system is twenty-fourth order which exceeds MIMO's tenth order capability. However, the actual coding of the algorithm is identical.

In addition to the application of the Moore algorithm, MIMO provides options for singular value vs. sample time plots (see Section III), discretize the continuous time linear, time-invariant state space system, estimate the condition number with respect to inversion for a square matrix, obtain the discrete-time controllability, observability, and Hankel matrices, obtain the steady state controllability and observability grammians, obtain the frequency response for the balanced system, list the singular values and singular vectors of a matrix as well as obtain various special state coordinate systems (see Section III).

MIMO utilizes overlays which allow a modular design and save computer memory. MIMO is designed to allow the implementation of future options which utilize the data base provided by the program.

Organization of Thesis

The thesis is organized in the following manner:

(1) Background and algorithm development is presented in Section II.

(2) In Section III, the interactive program MIMO is explained and its options presented.

(3) Section IV compares several model order reduction techniques with the Moore algorithm for a third order SISO (single-input-single-output) system.

(4) In Section V, the application of the Moore algorithm to the B-52E flutter control problem is presented. At this point, comparisons are made between the Moore technique and the modified Schwendler and MacNeal technique (Ref 32) currently in use by AFFDL.

(5) Finally, the conclusions and recommendations (Section VI), concerning Moore's algorithm and its applicability to problems of finding reduced order models for the B-52E flutter control problem, are presented.

II. PRESENTATION OF THE ALGORITHM

Introduction

With the advent of digital computers, the ability to analyze large dimension, linear systems has been established. Unfortunately, even with the speed of digital computers, many linear systems possess such a large dimension that real-time computations are prohibited. The model order reduction problem is borne from this type of situation. Though many techniques exist, there is no widely accepted method that completely solves the model order reduction problem. Therefore, the need for an efficient algorithm persists.

The Moore technique takes a novel approach in that it addresses whether a state is close to being redundant as versus merely accepting a given state to be redundant or non-redundant. Transforming a system into a state coordinate system in which this state redundancy is clearly presented is a topic introduced by Kalman (Ref 19) in 1963. The Kalman theory provides a basis for the Moore approach.

The remainder of this section will be organized in the following manner. First, the basic model order reduction problem will be defined. Second, the Kalman theory of minimal realization will be discussed. Next, the relationships between the Kalman theory and the Moore algorithm will be presented. The Moore algorithm actually consists of four paths. Each path will be discussed in detail as well as when a particular path might be desirable will be presented. Many existing model order reduction procedures work well, but they don't provide a basis for determining the minimum order of a reduced order model for an "accurate" (user-defined) reproduction of the original system. The Moore algorithm does present such decision information. This will be discussed following the presentation of the four algorithm paths. Finally, the algorithm will be summarized and unique properties discussed.

Problem

For this study, only linear, time-invariant systems are considered, and in analysis of these systems, the following problem is common: given a system of high dimensionality described by n first order differential equations as:

$$\underline{\dot{x}} = A\underline{x} + B\underline{u} \qquad (1)$$

$$\underline{y} = C\underline{x} + D\underline{u} \qquad (2)$$

where $\underline{x}$ is an n-vector (n x 1)

$\underline{u}$ is a q-vector of inputs (q x 1)

$\underline{y}$ is a m-vector of outputs (m x 1)

(usually n>>m or q)

A is n x n

B is n x q

C is m x n

D is m x q

This form is known as the state variable representation of the system. A system of the form:

$$\underline{\dot{x}}_r = A_r\underline{x}_r + B_r\underline{u} \qquad (3)$$

$$\underline{y}_r = C_r\underline{x}_r + D\underline{u} \qquad (4)$$

where $\underline{x}_r$ is a r-vector and $r << n$, is desired such that this low order model preserves the "important" input-output properties of the original system (1) - (2). These important properties include frequency response, transient and steady state response and that the resulting reduced order model remain at least as controllable and as observable as the original system.

Kalman Theory of Minimal Realization

Kalman uses Gilbert's canonical decomposition procedure to transform a state variable coordinate system into a coordinate system in which the system is clearly divided into completely controllable-completely observable (c. c. -c. o.), completely controllable-completely unobservable (c. c. -u. o.), completely uncontrollable-completely observable (u. c. -c. o.), and completely uncontrollable-completely unobservable (u. c. -u. o.) portions. Kalman states (Ref 19:169) that a system is irreducible if it consists only of the c. c. -c. o. portion of the original system. The input-output or transfer function properties of a system can only give information about the c. c. -c. o. representation of a system. Another interesting result presented by Kalman is that a system is irreducible if the rank of the matrix of impulse response functions (the Hankel matrix) is of rank n. Unfortunately, however, the c.c.-c.o., c.c.-u.o., u.c.-c.o., and u.c.-u.o. subspaces cannot be determined accurately on a computer due to their numerical instability. Also the subspaces are highly state coordinate system dependent, meaning that for equivalent state space representations of a system, different subspaces exist. The Moore algorithm presents a way to yield a coordinate invariant, numerically stable method of obtaining these subspaces. In effect, the Moore algorithm provides "working subspaces" for the c.c.-c.o. subspace.

Presentation of the Moore Algorithm (Ref 24,25) (Appendix B)

The Moore algorithm finds a state space coordinate system which orders the states with respect to controllability and observability properties (i.e., most controllable-most observable states to the least controllable-least observable states). By stripping away the bottom states (i.e., the least controllable-least observable states) model order reduction is performed.

The controllability grammian (Equation 5) and the observability grammian (Equation 6) allow the c.c.-c.o. subspace to be numerically solved for

$$W_c^2(t) = e^{At}\left[\int_0^t e^{-A\tau}BB^Te^{-A^t\tau}d\tau\right]e^{A^tt} \qquad (5)$$

$$W_o^a\,(t) = e^{A^T t}\,[\int_0^t e^{-A^T \tau} C^T C e^{-A\tau}\, d\tau\,] e^{At} \qquad (6)$$

The controllable subspace is

$$X_c = \text{Im}\,(W_c^a(t)) \qquad (7)$$

The uncontrollable subspace is

$$X_{\bar{c}} = \text{Ker}\,(W_c^a\,(t)^T) = \text{Ker}\,(W_c^a\,(t)) \qquad (8)$$

The observable subspace is

$$X_o = \text{Im}\,(W_o^a\,(t)^T) = \text{Im}\,(W_o^a(t)) \qquad (9)$$

The unobservable subspace is

$$X_{\bar{o}} = \text{Ker}\,(W_o^a(t)) \qquad (10)$$

Differentiating equations 5 and 6 respectively yields

$$\dot{W}_c^a\,(t) = AW_c^a(t) + W_c^a\,(t)\,A^T + [BB^T] \qquad (11)$$

$$\dot{W}_o^a\,(t) = A^T\,W_o^a(t) + W_o^a A + [C^T C] \qquad (12)$$

Allowing $\dot{W}_c^a\,(t)$ and $\dot{W}_o^a\,(t)$ to be zero in the steady state produces the algebraic Lyapunov equations

$$O = A\,W_c^a\,(t) + W_c^a\,(t)\,A^T + [BB^T] \qquad (13)$$

$$O = A^T\,W_o^a\,(t) + W_o^a\,(t)\,A + [C^T C] \qquad (14)$$

By solving these equations for $W_c^a\,(t)$ and $W_o^a\,(t)$ yields the controllability and observability grammians respectively. By characterizing the null space (kernal) and range space (image) of these grammians, the c.c.-c.o. subspace is identified.

However, as mentioned previously, these subspaces (grammians) are state coordinate system dependent. Therefore the Moore algorithm transforms the system into the internally balanced state coordinate system, $\underline{X} = T\,\underline{X}'$, using the linear transformation matrix

$$T = U_o\, \Sigma_o^{-1}\, U_H\, \Sigma_H^{\frac{1}{2}} \quad \text{(Ref 25, Appendix B)} \tag{15}$$

where U_o = left singular vectors of the observability grammian (for more information on singular values, vectors see Appendix A, Ref 36)

Σ_o = diagonal matrix containing singular values of the observability grammian

U_H = left singular vectors of the H_{INF} matrix

$H_{INF} = \Sigma_o\, U_o^T\, U_c\, \Sigma_c$

Σ_H = diagonal matrix containing singular values of the H_{INF} matrix

U_c = left singular vectors of the controllability grammian

Σ_c = diagonal matrix containing singular values of the controllability grammian

The Moore algorithm will now be presented. Each of the four main paths will be explained.

A flow chart illustrating the Moore algorithm is found in Figure 1. The reader is encouraged to refer to References 24, 25 and to Appendix B for the in-depth development.

Path 1 - Impulse Balancing. The algorithm flow-charted in Figure 1 is the "impulse balancing" algorithm. This algorithm yields its best results (i. e., lowest order models) for a system when subjected to an impulse input. This is not to say that reduced order models obtained via the impulse balancing method when responding to other classes of inputs (step, sine, etc.) will not respond well. The point is, the best replication of the original system for the lowest order model will occur if the model and the original system are *responding to an impulse input.*

Path 2 - Step Balancing. The algorithm is now modified to balance according to a step input. When this is accomplished, the reduced order models will be best for a step input.

The impulse response function (assuming no feedforward term) is

$$\underline{Y}(t) = Ce^{At} B \qquad (16)$$

The impulse is the time derivative of the unit step function (Ref 28:65). Therefore, the step response is written

$$\underline{Y}(t) = C(\int_0^t e^{A\tau} d\tau)B \qquad (17)$$

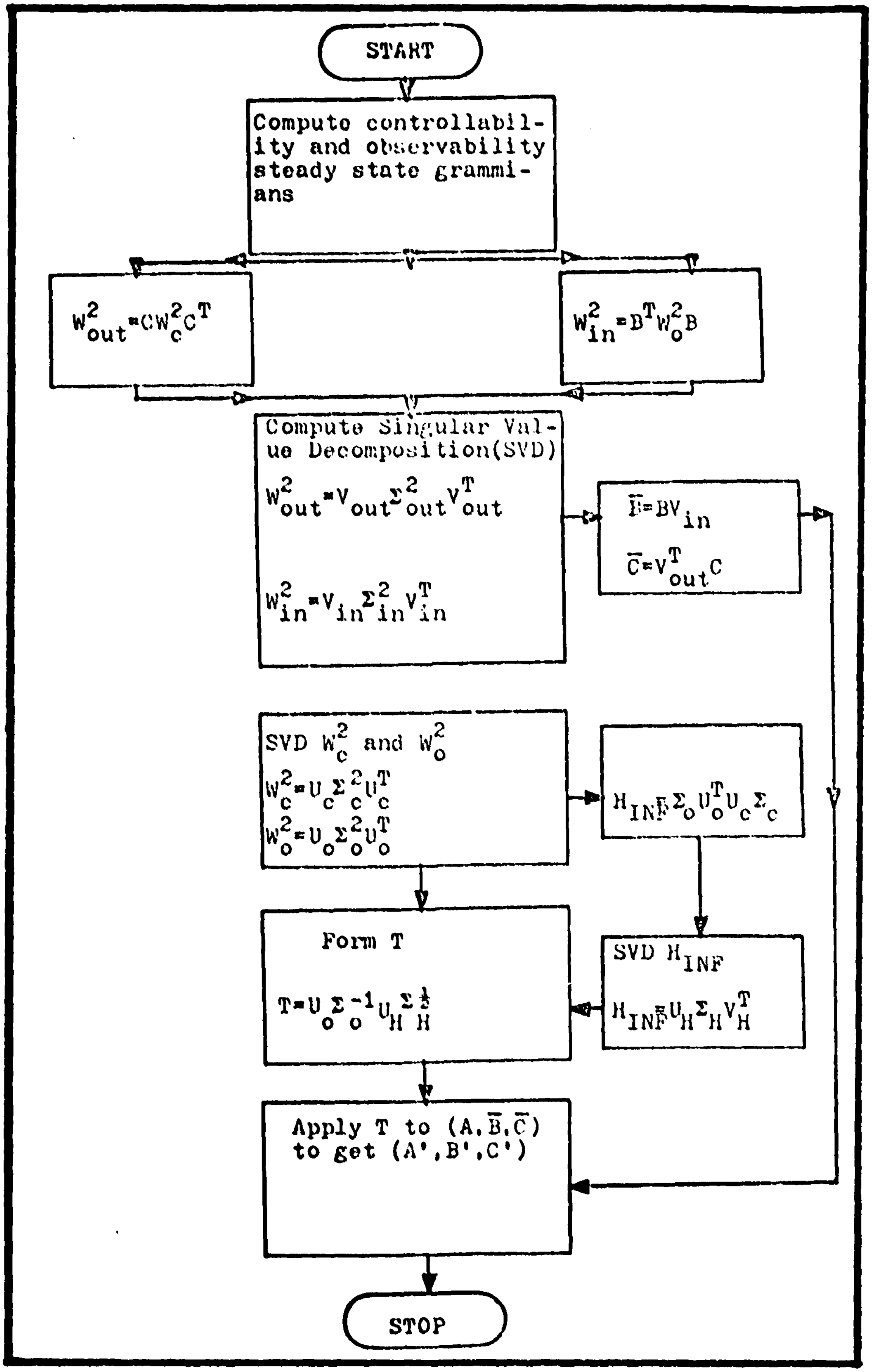

START
Compute controllability and observability steady state grammians
$W^2_{out}=CW^2_cC^T$
$W^2_{in}=B^TW^2_oB$
Compute Singular Value Decomposition(SVD)
$W^2_{out}=V_{out}\Sigma^2_{out}V^T_{out}$
$W^2_{in}=V_{in}\Sigma^2_{in}V^T_{in}$
$\bar{B}=BV_{in}$
$\bar{C}=V^T_{out}C$
SVD W^2_c and W^2_o
$W^2_c=U_c\Sigma^2_cU^T_c$
$W^2_o=U_o\Sigma^2_oU^T_o$
$H_{INF}=\Sigma_oU^T_oU_c\Sigma_c$
Form T
$T=U_o\Sigma^{-1}_oU_H\Sigma^{\frac{1}{2}}_H$
SVD H_{INF}
$H_{INF}=U_H\Sigma_HV^T_H$
Apply T to $(A,\bar{B},\bar{C})$ to get (A',B',C')
STOP

FIGURE 1. Flow Chart of Impulse Balancing Algorithm

Using the power series representation for $e^{A\tau}$ (Ref 29), equation (17) can be written

$$\underline{Y}(t) = C[e^{At}-I]A^{-1}B. \tag{18}$$

This is rearranged to produce

$$\underline{Y}(t) = Ce^{At}(A^{-1}B) - CA^{-1}B. \tag{19}$$

Using the state space representation of the transfer function matrix (Ref 11:131),

$$G(s) = C[sI-A]^{-1}B, \tag{20}$$

and assuming no feedforward elements (no D matrix), and zero initial conditions, yields

$$\underline{Y}(s) = C[sI-A]^{-1}B\underline{U}(s). \tag{21}$$

Applying the final theorem to equation (20) yields (Ref 22:103)

$$\underline{Y}(s)\Big|_{\substack{\text{steady}\\\text{state}}} = -CA^{-1}B. \tag{22}$$

Since the second term in equation (19) is indeed the steady state part of the step response, then the first term is the transient portion.

Therefore,

$$\underline{Y}(t)\Big|_{\substack{\text{transient}\\\text{(step input)}}} = Ce^{At}(A^{-1}B). \tag{23}$$

The Moore algorithm gives results for the impulse response

$$\underline{Y}(t) = Ce^{At}B. \tag{24}$$

Therefore, if (A, $A^{-1}B$, C) is input to the original balancing algorithm, it will yield the transient part of the step response rather than the impulse response. The steady state value differences between the reduced order model and the full order model are taken care of by adding a compensating D matrix to the reduced order system. This matrix is defined as

$$D_{COMP} = -[CA^{-1}B]_{FULL} - (-[CA^{-1}B]_{REDUCED}). \tag{25}$$

Using these two corrections, the algorithm now is a "step balancing" algorithm and utilizes the knowledge that the input is a step to provide lower order models for a step input than the impulse balancing algorithm can. Again, reduced order models providing "good" results can be obtained using both the impulse balancing and step balancing algorithms for other classes of inputs, but if the algorithm possesses knowledge of the type input, lower order models can usually be attained.

Path 3 - Infinite Interval Grammians.

Equations 13 and 14 presented the two matrix Lyapunov equations that yield the grammians for the interval between zero and infinity. This tends to weight the steady state characteristics of the balanced design more heavily than the transient characteristics.

Path 4 - Finite Interval Grammians.

If the original integrals in equations 5 and 6 are numerically integrated from 0 to t, where t is some finite time, the steady state characteristics of the resulting model would not be as dominant as before. This path is not investigated in this thesis. However, through experimentation an "optimal" t could probably be found to "weight" the balanced model characteristics equally. This path is discussed further in Section VI in the Conclusions and Recommendations.

The H_{INF} Matrix

Most model order reduction techniques do not present a means for determining the lowest possible order that the model can be reduced to, thereby making the user perform many iterations to find this minimum order. The Moore algorithm does offer a guideline. The indicator stems from the H_{INF} matrix which is an integral part of the Moore balancing algorithm. (See equation (15) and the definition of terms following that equation.) The H_{INF} matrix is related to the discrete-time Hankel matrix (matrix of impulse responses) (Ref 16), where the Hankel matrix may be defined as the discrete time observability matrix multiplied by the discrete time controllability matrix.

$$H_{ANKEL} = \begin{bmatrix} C \\ CF \\ \cdot \\ \cdot \\ \cdot \\ CF^{n-1} \end{bmatrix} [B\ FB \ldots F^{n-1}B] = \begin{bmatrix} CB\ CFB \ldots CF^{n-1}B \\ CFB \\ \cdot \\ \cdot \\ \cdot \\ CF^{n-1}B \ldots CF^{2(n-1)}B \end{bmatrix} \quad (26)$$

where $F = e^{AT}$ (T = sample time). The Hankel matrix is also identical to the matrix of impulse responses:

$$H_{ANKEL} = \begin{bmatrix} h(1) \ldots & h(N) \\ h(2) & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ h(n) \ldots & h(2N-1) \end{bmatrix} \quad (27)$$

Then, taking the singular value factorization of the H_{INF} matrix and Hankel matrix, respectively, yields the following result:

$$\sigma_{H^i_{INF}} = \lim_{T \to o} \sigma_{H^i_{ANKEL}} \quad (28)$$

where T is sampling time

σ = singular value (eigenvalue of $H_{INF}{}^T H_{INF}$)

Kalman in Reference 19 shows that the rank of the Hankel matrix (matrix of impulse response functions) gives information on whether a system is of minimum order. Moore uses the continuous-time, coordinate-invariant H_{INF} matrix to provide similar information.

This can best be understood by using the geometrical concept provided by the singular value factorization of the H_{INF} matrix (see Appendix A, B). The singular values of the H_{INF} matrix are the axis' lengths of the hyperellipsoid described by singular vectors of the H_{INF} matrix. The H_{INF} matrix (similar to the discrete-time Hankel matrix) has embedded in it controllability/observability properties. The "large" singular values are the axis' lengths of the major axes of the hyperellipsoid. These major axes span the controllable/observable subspace. Therefore, the "small" singular values correspond to the minor axes, and thus the uncontrollable/unobservable subspace. State redundancy is caused by uncontrollable/unobservable states (Ref 25). Therefore, the number of "large" singular values define the dimension of the controllable/observable (thus nonredundant) state space.

<u>Example</u>

An example is now presented which illustrates the state coordinate dependency of the controllability and observability grammians. The internally balanced representation will then be presented showing the coordinate invariant grammians (thus coordinate invariant subspaces).

The example has transfer function

$$G(s) = \frac{1}{(s+1+j2)(s+1-j2)(s+3)} \tag{29}$$

Three test cases were run. In each case the A matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15 & -11 & -5 \end{bmatrix} \tag{30}$$

For case 1, the B and C matrices are

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C_1 = [1\ 0\ 0] \tag{31}$$

For case 2, the B and C matrices are

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \times 10^{-2} \end{bmatrix}, \quad C_2 = [1 \times 10^{+2}\ 0\ 0] \tag{32}$$

For case 3, the B and C matrices are

$$B_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \times 10^{+2} \end{bmatrix}, \quad C_3 = [1 \times 10^{-2}\ 0\ 0] \tag{33}$$

The impulse balanced system for impulse input for each case is

$$A' = \begin{bmatrix} .358 & .508 & 1.71 \\ .508 & -3.46 & 1.61 \\ 1.71 & -1.61 & -1.18 \end{bmatrix} \tag{34}$$

$$B' \quad \begin{bmatrix} .202 \\ -.152 \\ .253 \end{bmatrix}, \quad C' = [.202 \quad -.152 \quad -.253]$$

The step balanced system (Moore balanced system for step input) for each case is

$$A' = \begin{bmatrix} 1.19 & .596 & 1.73 \\ .596 & -2.87 & 1.06 \\ -1.73 & -1.06 & -.941 \end{bmatrix} \qquad (35)$$

$$B' = \begin{bmatrix} .137 \\ -.532 \\ .550 \end{bmatrix}, \quad C' = [.286 \quad -.074 \quad -.142]$$

The respective controllability and observability grammians are shown in Table I. This table shows the considerable variance in the magnitude of the elements in the controllability and observability grammians for Cases 1, 2, 3. The grammians shown for the impulse balancing and the step balancing algorithms are unique for the given system.

TABLE I. Test Case Controllability and Observability Grammians

CASE	CONTROLLABILITY GRAMMIAN	OBSERVABILITY GRAMMIAN
1	$\begin{bmatrix} 4.2E-3 & 1.8E-12 & -1.3E-2 \\ 1.8E-12 & 1.3E-2 & 1.6E-11 \\ -1.3E-2 & 1.6E-11 & 1.4E-1 \end{bmatrix}$	$\begin{bmatrix} 6.8E-1 & 2.3E-1 & 3.3E-3 \\ 2.3E-1 & 1.2E-1 & 2.1E-2 \\ 3.3E-2 & 2.1E-2 & 4.2E-3 \end{bmatrix}$
2	$\begin{bmatrix} 4.2E-7 & 1.8E-16 & -1.3E-6 \\ 1.8E-16 & 1.3E-6 & 1.6E-15 \\ -1.3E-6 & 1.6E-15 & 1.4E-5 \end{bmatrix}$	$\begin{bmatrix} 6.8E+3 & 2.3E+3 & 3.3E+2 \\ 2.3E+3 & 1.2E+3 & 2.1E+2 \\ 3.3E+2 & 2.1E+2 & 4.2E+1 \end{bmatrix}$
3	$\begin{bmatrix} 4.2E+1 & 1.8E-8 & -1.3E+2 \\ 1.8E-8 & 1.3E+2 & 1.6E-7 \\ -1.3E+2 & 1.6E-7 & 1.4E+3 \end{bmatrix}$	$\begin{bmatrix} 6.8E-5 & 2.3E-5 & 3.3E-6 \\ 2.3E-5 & 1.2E-5 & 2.1E-6 \\ 3.3E-6 & 2.1E-6 & 4.2E-7 \end{bmatrix}$
Impulse Balancing	$\begin{bmatrix} 5.7E-2 & 1.3E-7 & -3.7E-7 \\ 1.3E-7 & 3.3E-3 & 3.0E-7 \\ -3.7E-7 & 3.0E-7 & 2.7E-2 \end{bmatrix}$	$\begin{bmatrix} 5.7E-2 & 8.2E-7 & 2.0E-7 \\ 8.2E-7 & 3.3E-3 & -7.4E-7 \\ 2.0E-7 & -7.4E-7 & 2.7E-2 \end{bmatrix}$
Step Balancing	$\begin{bmatrix} 8.1E-2 & -6.1E-3 & 5.2E-2 \\ -6.1E-3 & 3.2E-2 & -4.3E-2 \\ 5.2E-2 & -4.3E-2 & 1.1E-1 \end{bmatrix}$	$\begin{bmatrix} 3.4E-2 & 6.2E-8 & 3.8E-8 \\ 6.2E-8 & 9.4E-4 & -1.4E-7 \\ 3.8E-8 & -1.4E-7 & 1.1E-2 \end{bmatrix}$

The singular values of the H_{INF} matrix for impulse balancing are

$$\begin{aligned} \sigma_1 &= 5.7226 \text{ X } 10^{-2} \\ \sigma_2 &= 2.7235 \text{ X } 10^{-2} \\ \sigma_3 &= 3.3416 \text{ X } 10^{-3} \end{aligned} \tag{36}$$

The singular values of the H_{INF} matrix for step balancing are

$$\begin{aligned} \sigma_1 &= 3.4278 \text{ X } 10^{-2} \\ \sigma_2 &= 1.0774 \text{ X } 10^{-2} \\ \sigma_3 &= 9.4013 \text{ X } 10^{-4} \end{aligned} \tag{37}$$

As mentioned, these singular values provide a guideline for how low the system order can be reduced. For both the impulse balancing and step balancing paths, the singular values of the H_{INF} matrix seem to cluster. There seems to be two "large" singular values in each case. This indicates that this system probably should not be reduced below second order for an "accurate" representation of the original system.

The H_{INF} matrix coupled with the four paths in the Moore algorithm provide options that other model order reduction techniques do not possess.

Summary of the Moore Algorithm and Internally Balanced Coordinate System

The Moore algorithm is implemented with numerically stable code. The code uses readily available subroutine packages such as IMSL, EISPACK, and LINPACK (Ref 13).

Because of this numerically stable code, and because of the linear transformation (Equation 15) to a coordinate invariant state representation, the Moore algorithm provides "working subspaces" for the Kalman decomposition.

The resulting internally balanced state coordinate system orders the states with respect to controllability and observability properties. Therefore, deletion of the bottom states of the internally balanced representation is in effect stripping away the uncontrollable, unobservable (thus redundant) subspace.

The H_{INF} matrix provides a guideline for the selection of the minimum order for the resulting reduced order model. This feature does not exist with most other model order reduction techniques.

The algorithm also minimizes the maximum of the condition number of the controllability and observability grammians. This property indicates that the balanced system is more robust to perturbations in system parameters.

The algorithm has now been presented. The next step is to evaluate the algorithm by applying it to real world systems. But before this can be accomplished, a tool must be developed which possesses the capabilities to investigate the reduced order model in both the time domain and frequency domain. In addition, it will be desirable to investigate the singular values of the H_{INF} matrix and the controllability and observability grammians. The next section presents an interactive software package which accomplishes the above analysis objectives.

III. DEVELOPMENT OF INTERACTIVE SOFTWARE INTRODUCTION

INTRODUCTION

In this section, an interactive computer program is developed that provides options useful in the study of multi-input-multi-output systems. The program is appropriately called MIMO. MIMO was developed in addition and in parallel to the application of the Moore algorithm presented in the previous sections.

Several options incorporated in MIMO are direct results from the study performed on the Moore algorithm. The remaining options yield various "special" state coordinate systems, each of which possesses special properties. A user's manual for MIMO is contained in Appendix C. Appendix D contains the Fortran listing of MIMO.

The structure of the program MIMO will be presented first. A description of the functions performed by each option will follow. The reader is encouraged to refer to the user's manual for an explanation on how to access and use MIMO, for that information will not be discussed here.

PROGRAM STRUCTURE

Several objectives influenced the design of MIMO. MIMO is an interactive program and is restricted to 60K (octal) words of memory. The program is designed to be readily modified. Individual options may be easily added, deleted or changed. MIMO interfaces to TOTAL (Ref 21) for time domain and frequency domain responses to be obtained. In the user's manual, efforts are made to explain how to prevent premature termination of MIMO due to an input error. However, time precludes the incorporation of an extensive error checking capability into the program itself. The main emphasis is instead to provide the user with accurate, efficient options and not at this time with an "optimized" user interface.

MIMO utilizes overlays (Ref 8,10) to meet the memory restrictions and to provide a degree of modularity to its design. Overlays divide a large program into a set of smaller programs, each of which is loaded into memory if and when it is required. The loading of these smaller programs is controlled by an executive overlay which remains in memory at all times. The smaller programs which may be referred to as primary overlays, may be subdivided into secondary overlays if necessary. Thus, the largest amount of memory ever required is the sum of the executive overlay's memory requirement and the sum of the largest primary and secondary overlays' memory requirements. Since only one primary overlay can reside in memory with the executive overlay at any given time, the modularity of the program is achieved. Additional options become additional primary overlays. Any sub-functions peculiar to any one option may be made secondary overlays. A more extensive discussion of overlays is found in Reference 8.

The additional objective, that MIMO interface to TOTAL (Ref 21), is met by storing all pertinent information in mass storage on a file named "MEMAUX". TOTAL also uses a file named MEMAUX. By writing the appropriate information with the appropriate index, the interface is achieved. The exact procedure for accomplishing this objective is found under Option 12 in Appendix C.

MIMO allows the user to store all current, pertinent information in the local file "MEMORE". This allows the user to terminate MIMO, execute other interactive computer functions, return to MIMO, and recover to the point of departure from the program.

MIMO contains two options which produce plots. A high degree of flexibility with respect to plots is achieved by MIMO. The user may dispose his accumulated plots to the AFIT terminal automatically with Option 9. He may also perform the standard routes, save the plot file, or use TEKPLOT or CCPREV (see AFIT OCR). Extensive descriptions of these procedures may be found in the user's manual in Appendix C.

With these objectives as well as restrictions incorporated, MIMO's options were developed. The description of each option to the extent that it does not duplicate the information contained in the user's manual is now presented.

PRESENTATION OF OPTIONS

MIMO's options work exclusively with matrices. Correspondingly, only the state variable description of a dynamic system is utilized. To meet computer memory restrictions, matrices are limited to 10 x 10's. The controllability matrix is limited to a 10 x 30, the observability matrix to a 30 x 10, and the Hankel matrix to a 30 x 30. At this time, sixteen options exist in MIMO. Their descriptions follow.

Option 0 lists the available options. The list will be amended as additional options are added.

Option 1 causes MIMO to terminate. All pertinent information is automatically saved in the local file MEMORE.

Input of the state space matrices (A,B,C) is accomplished in the first part of Option 2. After input is accomplished, the user may cease execution of Option 2, or obtain the "impulse balanced", or the "step balanced" systems obtained via the Moore algorithm. Using these balanced systems, reduced order models are obtained by deleting the bottom states (see Appendix C, Option 2) (Ref 24,25).

Option 3 uses a truncated power series approximation to obtain the (F,G,C) discrete-time system for a given value of sample time, from the (A,B,C) system input in Option 2 (valid for a linear, time invariant system only). The discrete-time controllability, observability, and Hankel matrices (Ref 28) are also obtained. The user may optionally suppress the listing. This is useful, i.e., if the user wishes to obtain the Hankel matrix and proceed to Option 7 to obtain its singular values/singular vectors.

The power series used in Option 3 is now presented. For a linear time invariant system,

$$F = e^{AT} \text{ (where } T = \text{sampling time).} \tag{38}$$

The G matrix is defined by

$$G = \int_0^t e^{A\tau} d\tau B \tag{39}$$

The power series representation for e^{AT} is (Ref 28:37),

$$e^{AT} = I + AT + \frac{A^2T^2}{2!} + \ldots. \tag{40}$$

The power series representation for $\int_0^t e^{A\tau} d\tau$ is

$$\int_0^t e^{A\tau} d\tau = I T + \frac{AT^2}{2!} + \ldots \tag{41}$$

Therefore, avoiding having to take A^{-1}, e^{AT} may be represented as

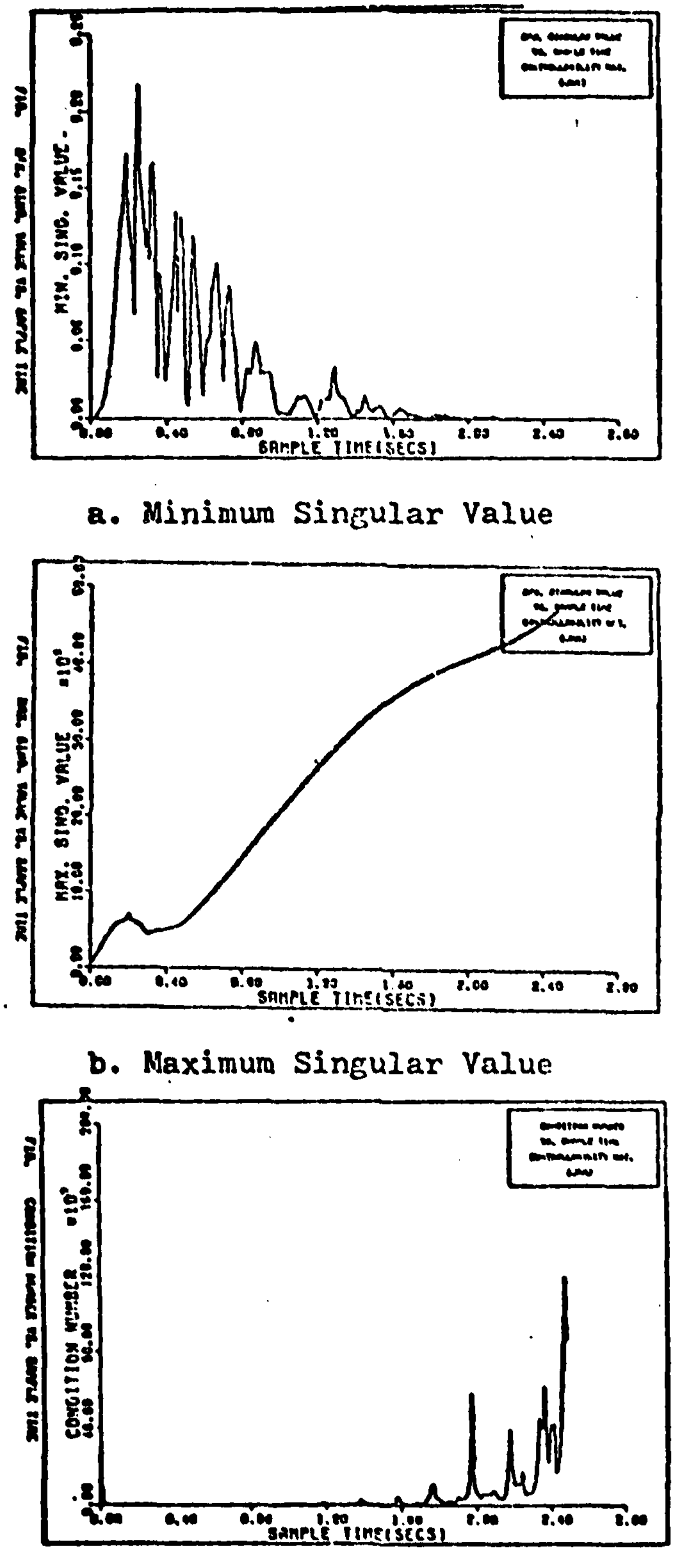
$$e^{AT} = A * [\int_0^t e^{A\tau} d\tau] + I \tag{42}$$

MIMO truncates this power series after 50 terms. The approximation is accurate if the magnitude of the largest eigenvalue of A multiplied by sample time is less than one (Ref 28:35). If this condition is not satisfied, convergence of the power series may not occur.

Option 4 plots and optionally lists singular values of the discrete time controllability, observability and/or Hankel matrices versus sample time. The user may list all singular values for each matrix for each sample time in a user-specified range, or he may just plot the singular values. The reader should refer to the user's manual for information on how to accomplish this option.

This option has several applications. The plot of the condition number ($\sigma_{max}/\sigma_{min}$) versus sample time, of the Hankel matrix provides an excellent criterion for choice of sample time for a discrete-time system (Ref 30). Plots of the minimum singular values of the controllability and observability matrices provide a measure of the degree of control, or degree to which a system may be observed respectively.

An example of this option as applied to the B-52E flutter control problem's twenty-fourth order system is now presented. All of the nine possible plots were generated. The B-52E model reveals that for the dominant eigenvalues, the period is .4051 seconds. Reid (Ref 30) shows that any integer multiple of this period would be a poor choice of sample time for a discrete time representation of the continuous-time system. The plots verify this fact. Figure 2 shows that the minimum singular value tends toward zero at integer multiples of .4051. As this singular value approaches zero, the



a. Minimum Singular Value

b. Maximum Singular Value

c. Condition Number (sig(max)/sig(min))

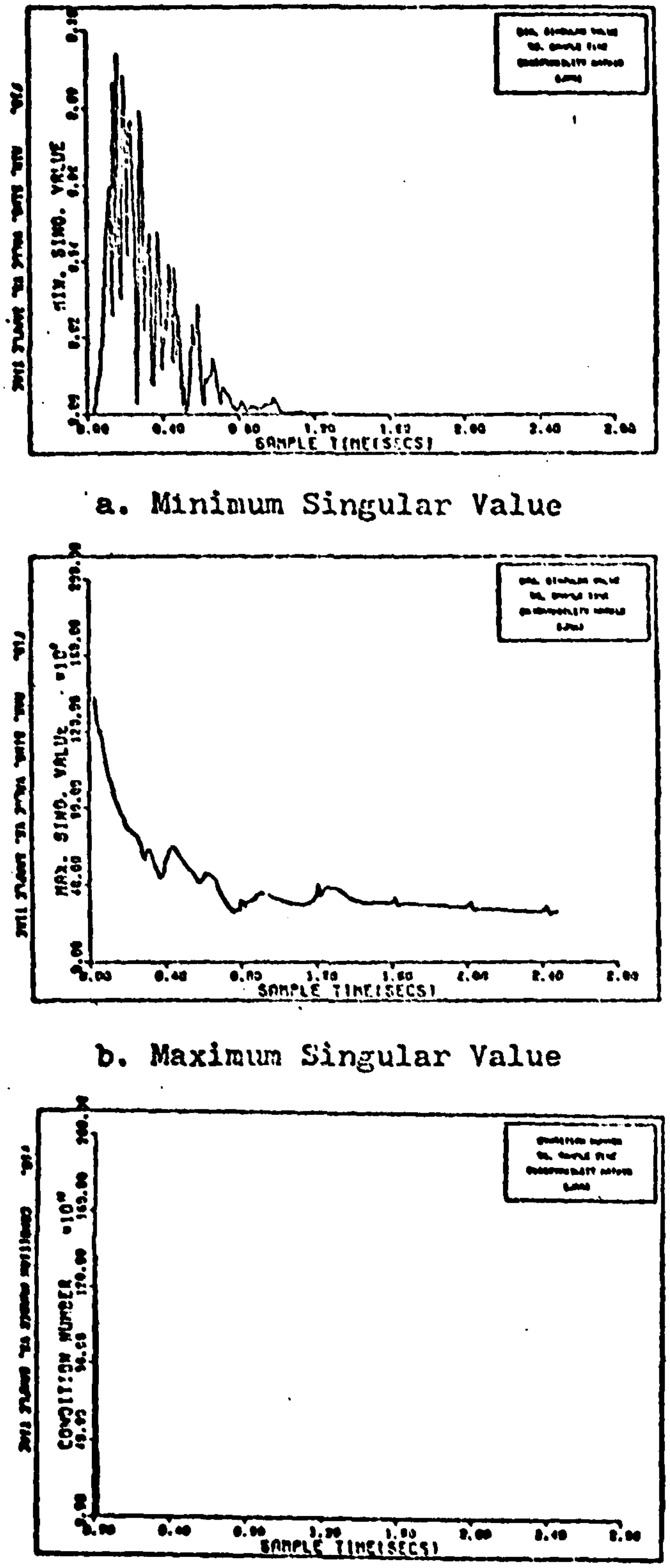
FIGURE 2. Controllability Matrix Max.and Min. Singular Values and Condition Number vs. Sample Time

controllability matrix is "closer" to singularity. This indicates that for an integer multiple of .4051, the degree of control on the system is lessened. Figure 3a shows that the observability matrix minimum singular value also tends toward zero as the sample time approaches an integer multiple of .4051. This indicates that the system is less observable at these sample times.

Perhaps though, Figure 4c is the most important. Clearly, the condition number of the Hankel matrix is maximized at integer multiples of .4051. This indicates that the system is most susceptible to perturbations (model mismatch/truncation, wordlength restrictions, etc.) in system elements. A "good" choice of sample time for a system then would be one that minimizes the condition number (K) of the Hankel matrix (maximized the reciprocal condition number in Reid's papers).

This option provides very revealing and helpful information in the choice of sample time for a system. For more information, see Reference 30.

Option 5 obtains the continuous time controllability and observability matrices for the (A,B,C) system input in Option 2. This option is useful if one desires to check the controllability and/or observability of his particular representation of a system. He may first input the system in Option 2, proceed to Option 5 to obtain these

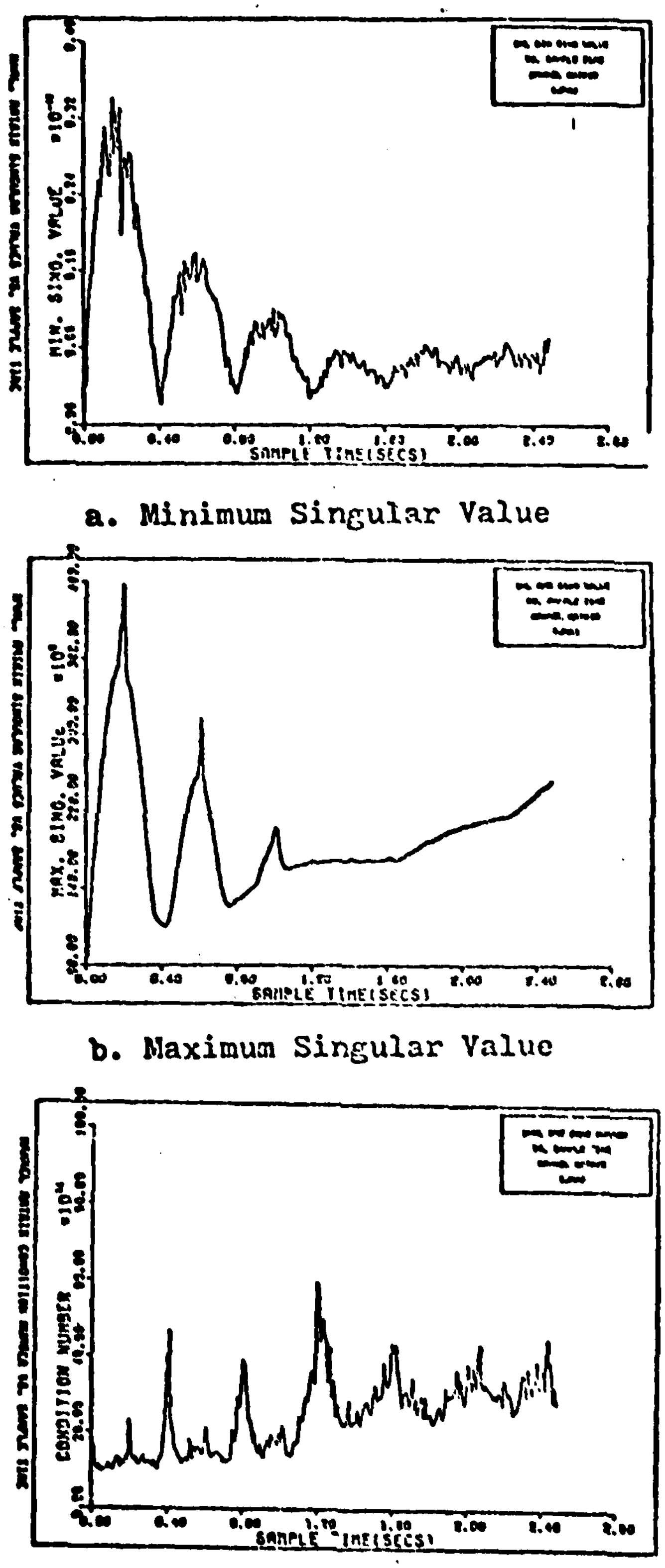


a. Minimum Singular Value

b. Maximum Singular Value

c. Condition Number (sig(max)/sig(min))

FIGURE 3. Observability Matrix Max. and Min. Singular Values and Condition Number vs. Sample Time



a. Minimum Singular Value

b. Maximum Singular Value

c. Condition Number (sig(max)/sig(min))

FIGURE 4. Hankel Matrix Max. and Min. Singular Values and Condition Number vs. Sample Time

matrices, and then go to Option 7 to obtain the singular values of the controllability and/or observability matrices. Generally, any singular value of magnitude less than 1.0×10^{-5} can be interpreted as zero. Therefore, the rank of the controllability and/or observability matrices is determined by the number of non-zero singular values.

Option 6 implements a portion of LINPACK (Ref 13) to estimate the condition number of a square, real matrix. This method as developed in Reference 7 is a highly efficient method for obtaining the condition number with respect to inversion. This method does not require the relatively expensive singular value decomposition. Instead, the condition number can also be defined as

$$K = \|A^{-1}\| \ \|A\| \tag{43}$$

where A is a real matrix

$\| \cdot \|$ denotes the subordinate spectral norm for matrix, i.e.,

$\| \dot{-} \|$ denotes the standard Euclidean (l_2) norm for a vector.

Using this definition, the algorithm yields a highly accurate estimate of the condition number with respect to inversion.

An example is now presented. One of the most ill-conditioned matrices with respect to inversion is a Hilbert matrix. A Hilbert matrix is of the form.

$$H_{IL} = \begin{bmatrix} 1/2 & 1/3 & \cdots & 1/N+1 \\ 1/3 & 1/4 & \cdots & 1/N+2 \\ \vdots & & & \vdots \\ 1/N+1 & \cdots & & 1/2N \end{bmatrix}_{NxN} \qquad (55)$$

As the order of the system increases, the estimate is more ill conditioned. In Table II the estimate of the condition number with respect to inversion is compared with $\sigma_{max}/\sigma_{min}$, obtained via the matrix's singular value decomposition. The results are quite good. Therefore, when the condition number of a matrix is desired, Option 6 is an efficient way of producing that result via the LINPACK algorithm (Ref 7, 13) for most applications.

TABLE II. Comparison of LINPACK Estimate of Condition Number Vs. $\sigma_{max}/\sigma_{min}$ for a Hilbert Matrix

Matrix Order	Option 6 Estimate	$\sigma_{max}/\sigma_{min}$	DIF
2	43.097	37.836	5.261
3	545.691	441.952	103.738
5	329.236	238.498	90.737

Option 7 obtains the singular values and optionally the left and right singular vectors for a real, rectangular matrix. The particular matrix can be: A matrix input at this time, the original A matrix, the internally balanced A matrix, the discrete time F matrix, or the discrete time Hankel matrix.

The singular value factorization of a real, rectangular matrix yields (see Appendix A, Ref 36):

$$A_{MXN} = U_{MXM} \Sigma_{MXP} V^T_{NXN} \tag{45}$$

where U is a matrix whose columns are called the left singular vectors.

Σ is a diagonal matrix whose diagonal elements are called the singular values and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq \sigma_{r+1} = 0$.

$r = \min (M, N)$

V is a matrix whose columns are called the right singular vectors.

To present the multiple uses of this factorization is beyond the scope of this thesis. The reader is referred to References 13, 24, 25, 28, 36, or 30 for excellent presentations on the subject.

Option 8 plots and optionally lists the magnitude and phase responses of a user specified reduced order system. The method used reduces the amount of computations as required by conventional algorithms and utilizes highly efficient eigenvalue - eigenvector routines to obtain its results.

The derivation follows (Ref 28):

$$G(jw) = F\{Ce^{At}B\} \qquad (46)$$

$$= F\{\sum_{i=1}^{N} C(\underline{V}j)(\underline{W}_j^{T} B)e^{\lambda_j t}\} \qquad (47)$$

where $F\{\cdot\}$ denotes the Fourier transform

$\left[\underline{V}_1\ \underline{V}_2 \cdots \underline{V}_n\right]$ = matrix of eigenvectors

$\left[\underline{W}_1\ \underline{W}_2 \cdots \underline{W}_n\right]^T = \left[\underline{V}_1\ \underline{V}_2 \cdots \underline{V}_n\right]^{-1}$

λ_j denotes the jth eigenvalue

Now, rewriting Equation 47 yields:

$$G(jw) = \sum_{i=1}^{N} (C\ \underline{V}i)(\underline{W}_i^T\ B)\ \frac{1}{jw + \lambda i} \qquad (48)$$

Using Equation 48 above, the magnitude and phase of the system for a given value of w can easily be obtained.

Option 9 disposes accumulated plots created by Option 4 and/or Option 8 to the AFIT terminal. This option uses an AFIT subroutine written by Captain Jerry Stinson (AFIT OCR). It also uses two subroutines contained in the BATELLE package (see AFIT OCR). The program can easily be modified to send the plots to any remote terminal with an on-line Calcomp plotter at Wright-Patterson Air Force Base, Ohio.

Option 10 writes all pertinent information to local file MEMORE. Option 10 is automatically called when Option 1 is executed.

Option 11 recovers all pertinent information from the local file MEMORE. This enables the user to recover to the point in MIMO where he used Option 1 or Option 10.

Option 12 interfaces to TOTAL (Ref 26). The original (A,B,C) system, and the internally balanced (A',B',C') system are written to the mass storage file MEMAUX. Appendix C, Option 12 explains how to access TOTAL and use the information transferred to TOTAL from MIMO. TOTAL (Ref 21) is an interactive program which produces time domain and frequency domain responses for a transfer function with a defined input. Numerous options useful to the control system designer are available. Therefore, the interface to TOTAL from MIMO saves the user from having to retype large matrices.

Option 13 generates the output normal state coordinate system (Ref 24, 25) from the internally balanced system obtained for the original input system. This system is a byproduct of the Moore algorithm. The system is now balanced with respect to observability properties only. The states are ordered from the most observable to the least observable. This option is limited to SISO systems only.

Option 14 produces the output predictive state coordinate system (Ref 30, 29). The transformation:

$$\underline{X} = M_{OBS}^{-1}\ \overline{\underline{X}} \tag{49}$$

where M_{OBS} is the observability matrix yields:

$$\overline{M}_{OBS} = I \tag{50}$$

Reid shows in Reference 30 that:

$$\underline{Y}\ (k+n/k-1) = \begin{bmatrix} C \\ CF \\ \vdots \\ CF^{n-1} \end{bmatrix} \tag{51}$$

where $\underline{Y}\ (k+n/k-1) = \begin{bmatrix} y\ (k+n/k-1) \\ y\ (k+n+1/k-1) \\ \vdots \\ y\ (k+2n-1/k-1) \end{bmatrix}$

where $\underline{Y}$ (k+n/k-1) denotes the vector of predicted outputs, to n samples in the future based upon inputs up to the current time k. By transforming to the output predictive state coordinate system, the following equation results

$$\underline{Y}\ (k+n/k-1) = \overline{F}^{n}\ \overline{\underline{X}}\ (k) \tag{52}$$

where $\overline{F}^n = M_o^{-1} F M_o$

M_o = observability matrix

$F = e^{AT}$

This implies that the outputs at discrete times in the future based on inputs up to time K may be determined by multiplying the current state vector $\underline{X}$ (K) by $\overline{F}^n$ (where $\overline{F}^n$ is now in the "output predictive" state coordinate system). See Reference 30 for uses of this coordinate system.

Finally, Option 15 obtains the steady state controllability and observability grammians for the (A,B,C) system input in Option 2. The controllability grammian is defined as:

$$W_c^a (O, t_1) = e^{At} \int_0^{t_1} e^{At} B B^T e^{A^T t} dt \quad e^{A^T t} \tag{53}$$

The observability grammian is defined as:

$$W_o^a (O, t_1) = e^{A^T t} \int_0^{t_1} e^{A^T t} C^T C e^{At} dt \quad e^{At} \tag{54}$$

Differentiating the two grammians respectively yields the two Ricatti equations

$$\dot{W}_c^a (t) = A W_c^a (t) + W_c^a(t) A^T + B B^T \tag{55}$$

$$\dot{W}_o^a (t) = A^T W_o^a (t) + W_o^a(t) A + C^T C \tag{56}$$

In the steady state, $\dot{W}_c^a$ and $\dot{W}_o^a$ equal zero. Therefore equations 55 and 56 become respectively:

$$A\,W_c^a(t) + W_c^a(t)\,A^T + BB^T = 0 \tag{57}$$

$$A^T W_o^a(t) + W_o^a(t)\,A + C^T C = 0 \tag{58}$$

Note, that the underlying (A,B,C) system must be stable because the modified Bartels-Stewart algorithm that solves the required matrix Ricatti equations must have negative eigenvalues to work with. This author believes newer algorithms exist that surmount this problem, but has not been able to locate such. The interested reader should refer to Reference 20 as the possible source of new algorithms that solve matrix Lyapunov equations.

This concludes the presentation of the interactive program MIMO. All goals strived for were met. The program is interactive (60K(octal) words of memory are required), uses overlays which provide a modularity and ease of modification to the program, interfaces to TOTAL (Ref 21) as desired, and although as yet doesn't possess an extensive error checking capability, does utilize the most efficient algorithms currently available to provide the user with a unique and hopefully beneficial analysis tool.

IV. APPLICATION OF MOORE ALGORITHM TO SISO EXAMPLE

Introduction

In Section II the Moore algorithm was presented and its potential for application to real world systems discussed. However, it was pointed out that an analysis tool would be required to investigate the actual performance of the Moore algorithm on real-world problems. Section III presented such a tool. MIMO coupled with TOTAL and some batch software now provide the means by which the Moore algorithm will be investigated. However, before the algorithm is applied to a large dimension, real world system, a third order SISO example will be presented which illustrates the application of the Moore algorithm, utilizes MIMO and TOTAL for analysis purposes, and compares the Moore algorithm with three other model reduction algorithms.

This section will be organized in the following manner. First, a brief discussion of the other three algorithms will be presented to familiarize the reader with the model order reduction approaches compared with Moore's algorithm for this example. Then, the third order example will be presented. A summary of the section will follow.

Description of Other Algorithms

Though the Moore algorithm works with the state space representation of a system, the three algorithms to be presented here use the system transfer function for their calculations. The transfer function matrix is related to state space (assuming no feedforward) via:

$$G(s) = C[sI-A]^{-1}B \tag{59}$$

T. Shamash (Ref 33) utilizes the Routh stability criterion and the Padé approximation to yield reduced order models. His basic procedure follows. Given:

$$G(s) = \frac{d_1 S^{n-1} + d_2 S^{n-2} + \ldots + d_n}{e_0 S^n + e_1 S^{n-2} + \ldots . + e_n} \tag{60}$$

compute:

$$\hat{G}(s) : \hat{G}(s) = \frac{1}{S} G\left[\frac{1}{S}\right] = \frac{d_n S^{n-1} + \ldots + d_1}{e_n S^n + e_{n-1} S^{n-2} + \ldots e_0} \tag{61}$$

Once this has been accomplished, $\hat{G}(S)$ is expanded into a continued fraction as:

$$\begin{aligned} \hat{G}(s) = {} & \frac{1}{1+\alpha_1 S} + \frac{1}{\alpha_2 S} + \ldots \frac{1}{\alpha_n S} \quad [B_1 \\ & + \frac{1}{\alpha_2 S} + \frac{1}{\alpha_3 S} + \ldots \frac{1}{\alpha_n S} \quad [B_2 \\ & + \\ & \quad \cdot \\ & \quad \cdot \\ & \quad \cdot \; \frac{1}{\alpha_n S} \; [B_n] \ldots]], \end{aligned} \tag{62}$$

where

$$\frac{1}{1+\alpha_1 S}+\frac{1}{\alpha_2 S}+\dots\frac{1}{\alpha_n S}=\cfrac{1}{1+\alpha_1 S+\cfrac{1}{\alpha_2 S+\cfrac{1}{\alpha_3 S+.\;\;\raisebox{-1em}{.}\;\;\raisebox{-2em}{.}\;\cfrac{1}{\alpha_n S}}}} \tag{63}$$

This procedure allows the algorithm to obtain the Routh coefficients from the α_i's and B_i's in Equations 62 and 63. Then, for the Kth order transfer function (where K << N), the first K terms are summed to yield R(S). The reciprocal relationship as used in Equation (61) is utilized to produce R(S), where R(S) is of lower order than the original G(S). This is a simplified presentation of the Shamash algorithm. In fact, there are many more computations used to efficiently obtain the coefficients of R(S).

The Shamash algorithm has many desirable properties. It preserves original system stability and can be applied to multivariable and discrete-time systems. However, it uses the transfer function f or its computations of the reduced order transfer function. For multivariable systems, especially, the storage requirements get to be huge. Thus the computational burden is increased by the use of transfer functions versus state space for a fairly large order, multivariable system.

El Attar and Vidyasagar (Ref 14) have developed an algorithm which treats the impulse response (or transfer function) as an input-output map. The procedure consists of minimizing one of two possible functionals. The choice of functional is dependent on the type of input used. In effect, the algorithm gives a best possible uniform approximation for the given input. This algorithm is useful when the poles of a system are not widely separated (i.e., a nondominant set of poles may not exist). In addition, original system stability is preserved and the algorithm is applicable to discrete-time systems, MIMO systems, and unstable systems. The algorithm, however, is by necessity input dependent.

Reddy (Ref 27) minimizes the integral of the square of the error between the corresponding real and imaginary parts of the original and assumed transfer functions' numerator and denominator to thereby evaluate the proper coefficients for the simplified transfer function. This algorithm is only applicable to SISO systems and does not ensure that the reduced order transfer function is stable, for a stable, original transfer function.

SISO Example

The following third order SISO example is presented as a comparison between the Moore algorithm and three other algorithms discussed in the previous section. The results of the model reduction for these other three algorithms are taken from Ref. 14. The results for the Moore algorithm are obtained via use of MIMO. The unit impulse responses and frequency responses are obtained for Moore's second order model, as well as for El-Attar's (Ref 14), Shamash's (Ref 33), and Reddy's (Ref 27) second order models.

The original third order transfer function is:

$$GH(s) = \frac{1}{(S+0.99)(S+1.0)(S+1.1)} \tag{64}$$

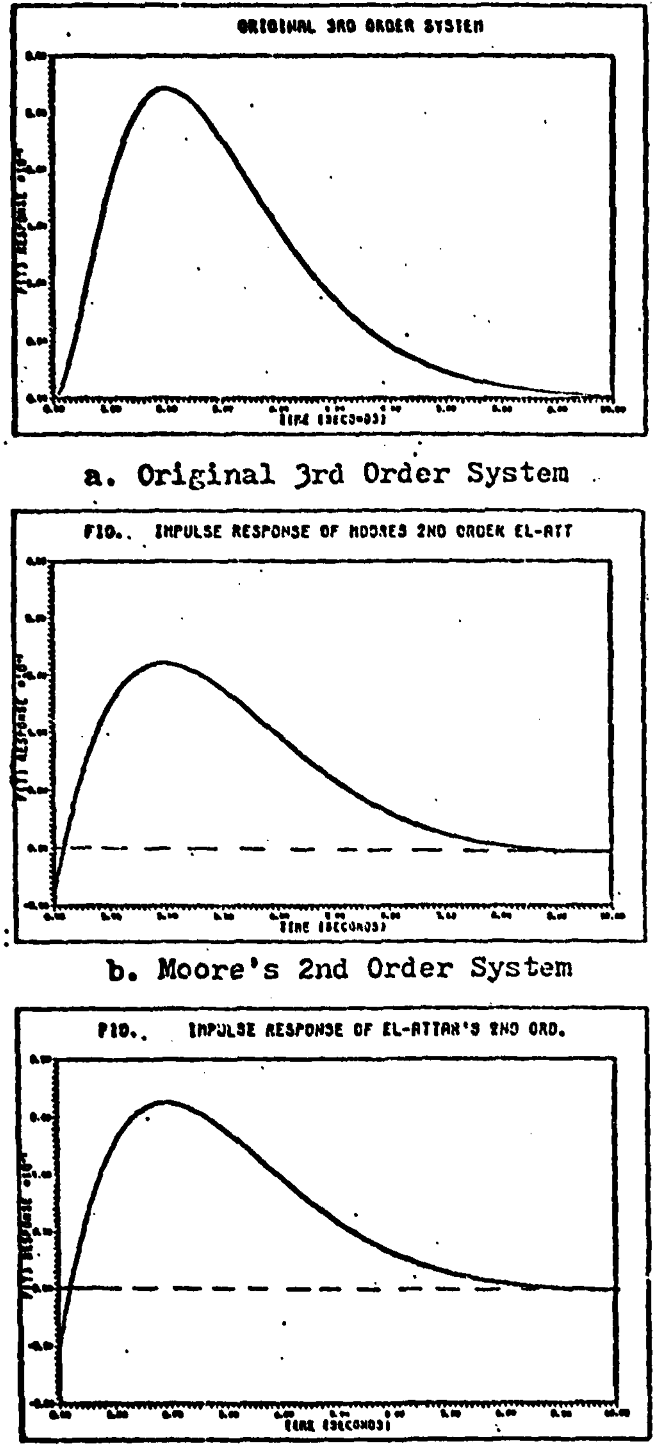
El Attar minimizes functions of induced norms. Shamash uses Pade' approximants and the Routh Stability criterion. Reddy minimizes an error criterion which is a function of the difference between the full order transfer function and the reduced order transfer function. Reddy's procedure results in an unstable second order model. The resulting reduced order transfer functions are shown in Table III.

The impulse response to a unit impulse input of the original system is shown in Figures 5a, 6a. Moore's second order system in response to the unit impulse is shown in Figure 5b. El-Attar's second order system in response to a unit impulse is in Figure 5c. El-Attar's and Moore's systems seem to be very close. This is true because El-Attar is using induced norm characteristics, while Moore's algorithm, based primarily upon the singular value factorization (where $\sigma_{max} = \|A\|$, A is a general matrix) (see Appendix A), is directly related to induced norms. Shamash's and Reddy's second order system's unit impulse response responses are found in Figures 6b and 6c, respectively.

TABLE III

Reduced Order Model Transfer Functions for SISO Example

ALGORITHM	TRANSFER FUNCTION
1. Moore	$\frac{-.4462s+.5870}{s^2+.09971s+.8827}$ $= \frac{-.4462(s-1.316)}{(s+.04985\pm j.9382)}$
2. El-Attar Vidyasagar	$\frac{-.0911s+.3806}{s^2+1.0054s+.3869}$ $= \frac{-.0911(s-4.1778)}{(s+.5027\pm j.3663)}$
3. Shamash	$\frac{.00022s+.3746}{s^2+1.12s+.3708}$ $= \frac{.00022(s+1702.73)}{(s+.56\pm j.2391)}$
4. Reddy	$\frac{1}{2.5s^2-1.34845s+1}$ $= \frac{.4}{(s-.620)\ (s+.080)}$

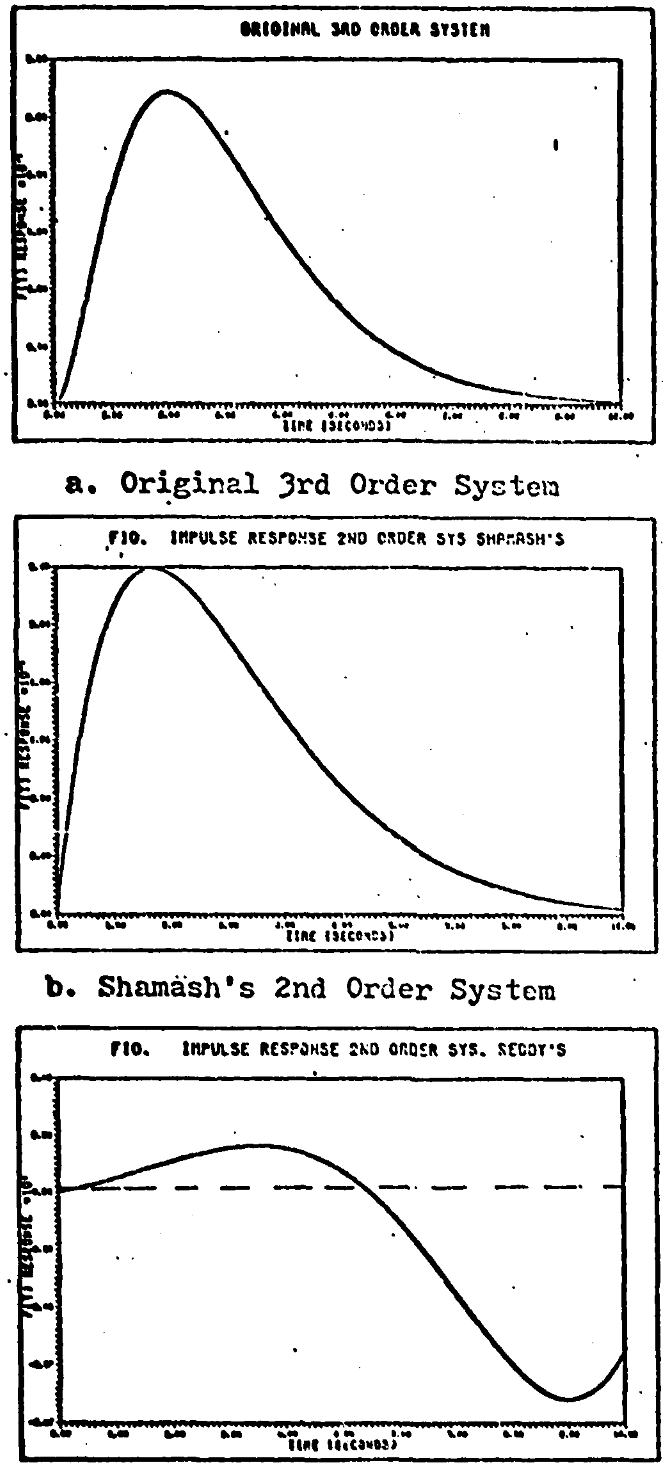


a. Original 3rd Order System

b. Moore's 2nd Order System

c. El-Attar's 2nd Order System

FIGURE 5. Unit Impulse Responses for SISO Example

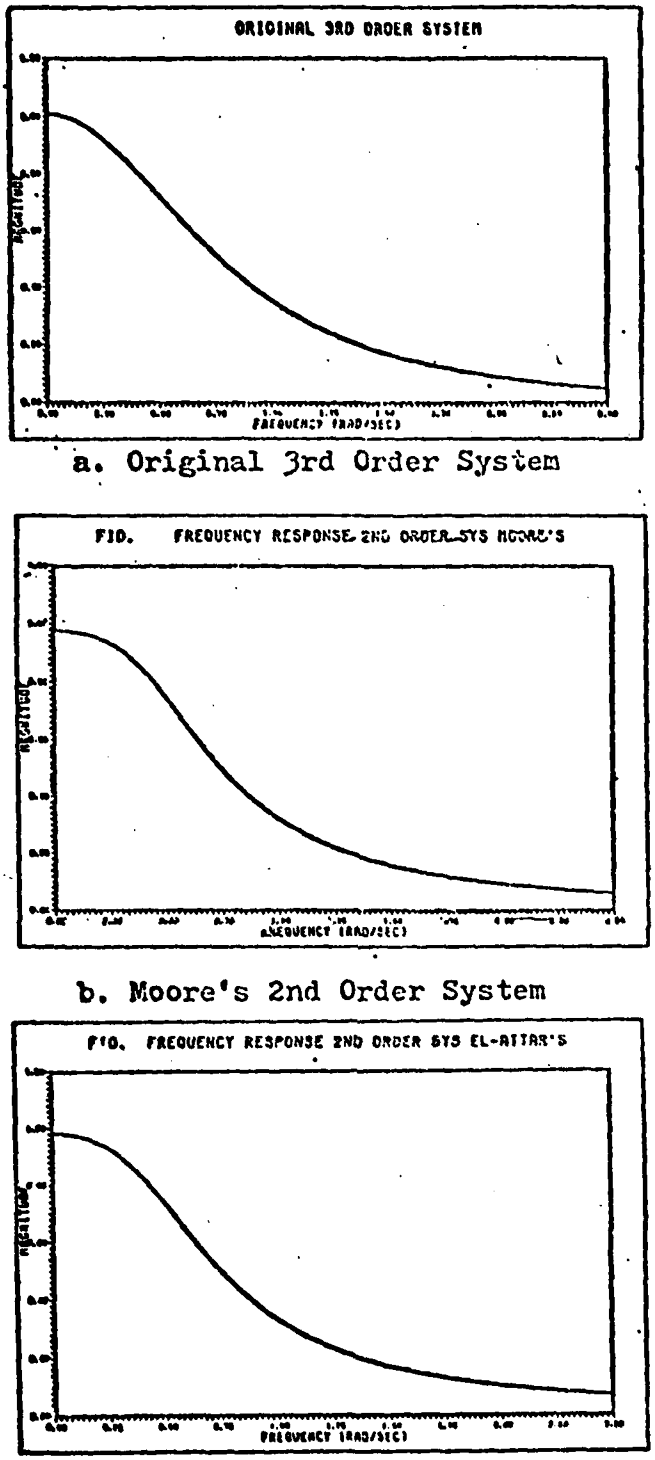


a. Original 3rd Order System

b. Shamash's 2nd Order System

c. Reddy's 2nd Order System

FIGURE 6. Impulse Responses for SISO Example



a. Original 3rd Order System

b. Moore's 2nd Order System

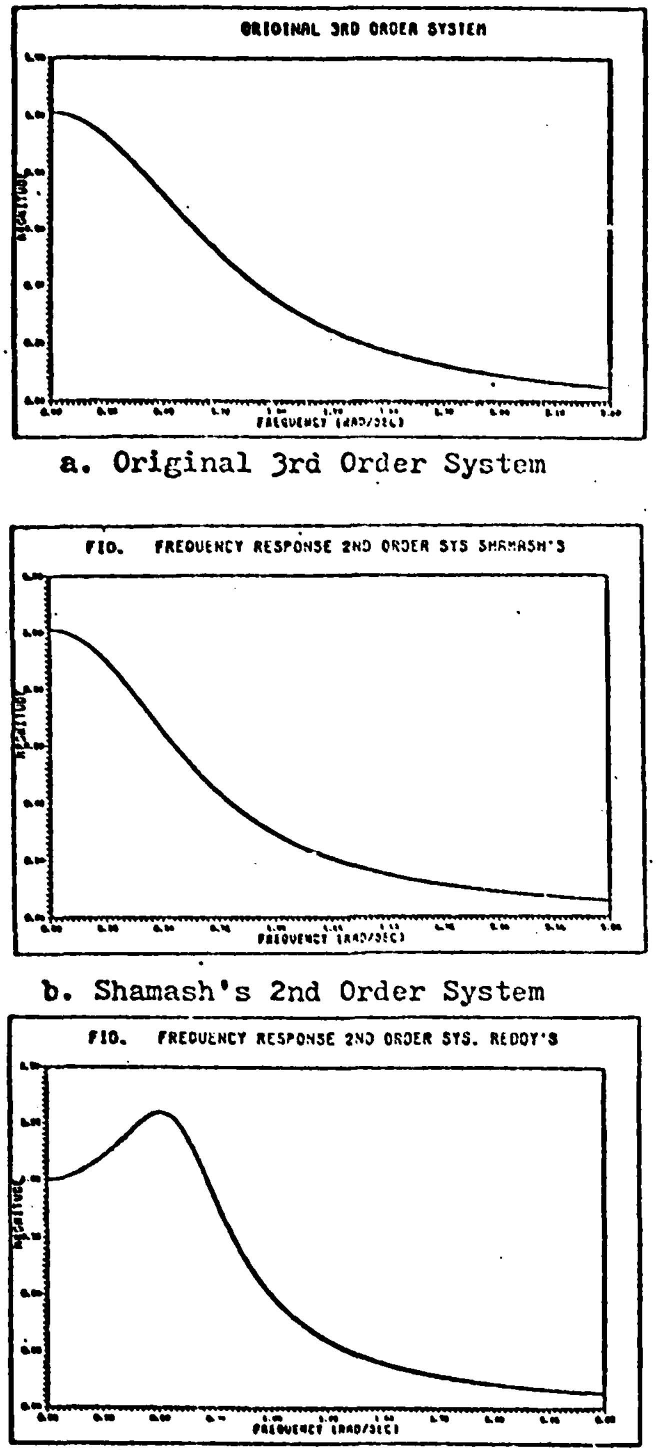
c. El-Attar's 2nd Order System

FIGURE 7. Magnitude Responses for SISO Example

The frequency response of the original third order system is found in Figures 7A, 8A, 9A, 10A. Moore's second order system frequency response (magnitude and phase) is found in Figures 7B, 9B. El-Attar's frequency response for his second order model is located in Figures 7C, 9C. Shamash's system is found in 8B, 10B. Reddy's unstable second order system's frequency response is in Figures 8C, 10C.

For the reduced order transfer functions that are shown in Table III, the Moore algorithm transfer function's dominant roots are the nearest to the origin and have the largest undamped natural frequency of all the algorithms. Yet the time domain and frequency domain responses are very close to those of El Attar and Vidyasagar. It should be noted that the Moore algorithm actually "shifts" its poles and zeros to compensate for the reduction of system order. Shamash's algorithm seems to have performed the best for this particular example. But as the order of the system increases, it becomes increasingly more difficult for a computer to store all of the transfer function coefficients. It is therefore more computationally efficient to work with the state space representation for a large dimension system (Ref 17).

The state space representation of the internally balanced (impulse balanced) system for this example is

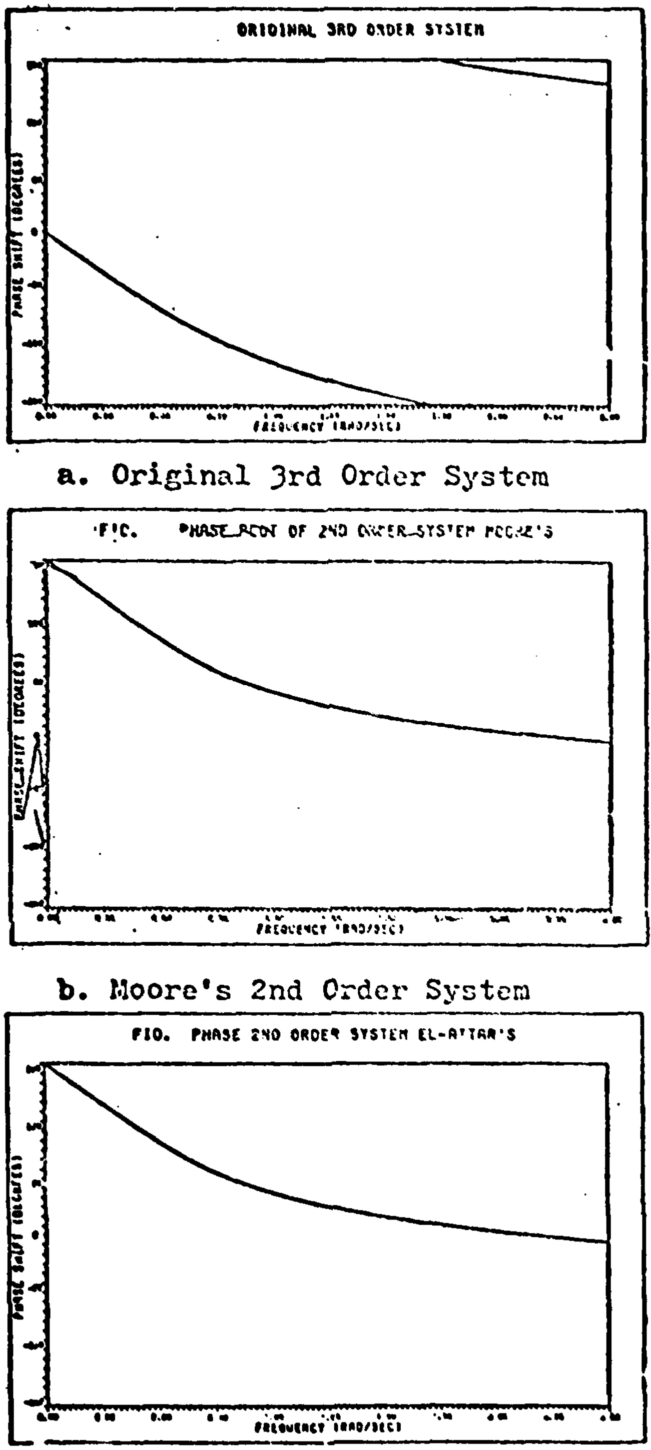


a. Original 3rd Order System

b. Shamash's 2nd Order System

c. Reddy's 2nd Order System

FIGURE 8. Magnitude Responses for SISO Example

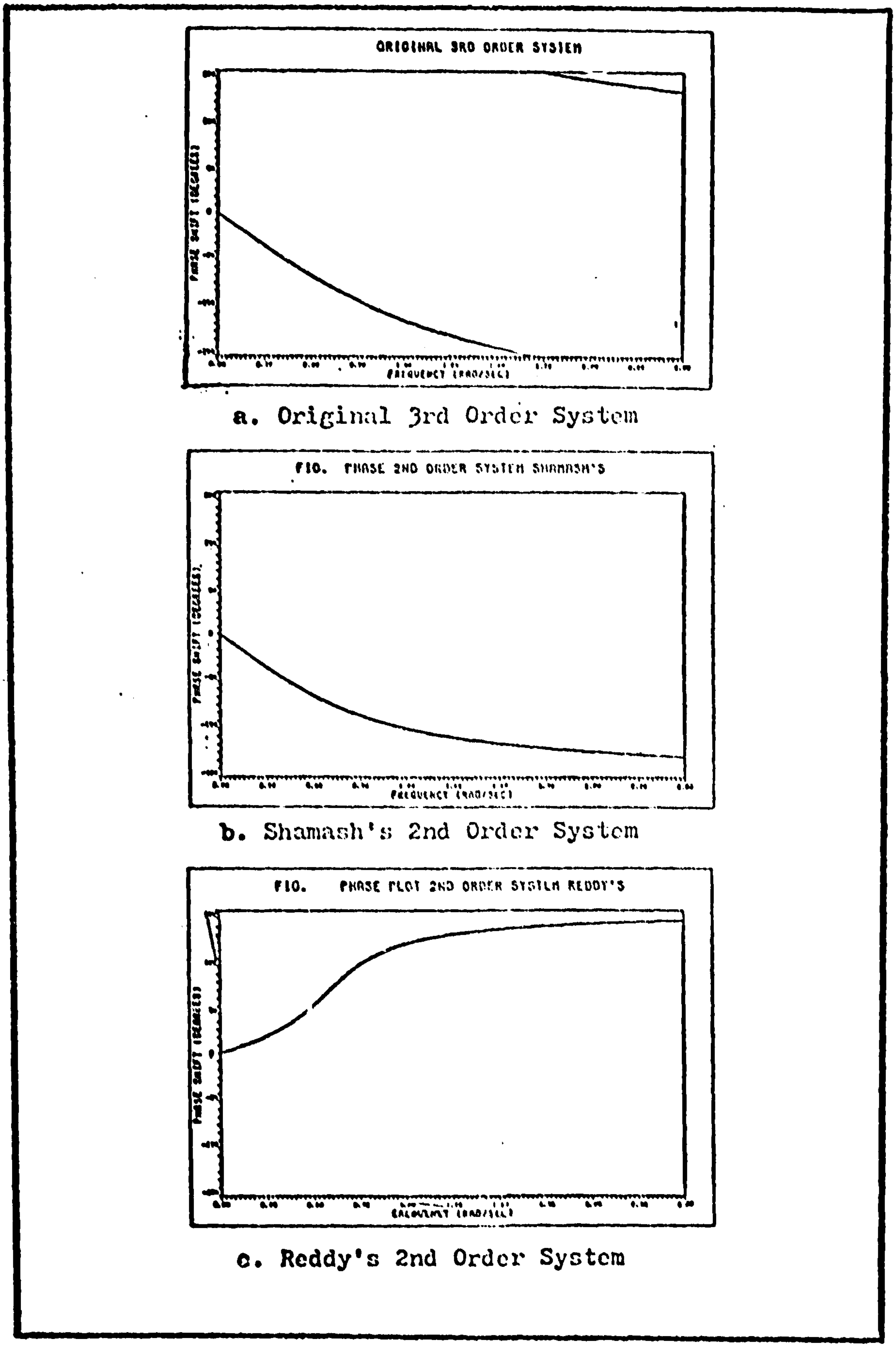


a. Original 3rd Order System

b. Moore's 2nd Order System

c. El-Attar's 2nd Order System

FIGURE 9. Phase Responses for SISO Example



a. Original 3rd Order System

b. Shamash's 2nd Order System

c. Reddy's 2nd Order System

FIGURE 10. Fhase Responses for SISO Example

$$A' = \begin{bmatrix} -8.02 \times 10^{-2} & 9.38 \times 10^{-1} & 1.46 \times 10^{-1} \\ -9.38 \times 10^{-1} & -1.95 \times 10^{-2} & 6.17 \times 10^{-2} \\ -1.46 \times 10^{-1} & 6.17 \times 10^{-2} & -1.00 \times 10^{0} \end{bmatrix} \qquad (65)$$

$$B' = \begin{bmatrix} 7.79 \times 10^{-1} \\ -4.00 \times 10^{-1} \\ 6.68 \times 10^{-1} \end{bmatrix} , \; C' = \begin{bmatrix} -7.79 \times 10^{-1} & -4.00 \times 10^{-1} & 6.68 \times 10^{-1} \end{bmatrix}$$

Notice that for a SISO system, there is an absolute value symmetry in the balanced A matrix for either step balancing or impulse balancing. In addition, for impulse balancing, as in this case, the balanced B and C matrices will be identical within a change of sign. This property provides a check for computational accuracy. The number of decimal places the symmetrical terms agree to give an indication of the accuracy to which the algorithm converged. This is often useful when the routine within MIMO that solves the two Lyapunov equations for the infinite interval controllability and observability grammians, does not converge satisfactorily. System order is reduced to two by deleting the third state (thus 3rd row of A', B', and 3rd column of C').

This low order SISO example was presented to tie together the algorithm and its properties with the tools that actually implement and provide analysis of the algorithm. However, the example is insufficient to illustrate fully the properties of the Moore algorithm.

Therefore the next section shall take a relatively large dimension real world system, apply the Moore algorithm to that system, and analyze and discuss the results of the application.

V. APPLICATION OF MOORE'S ALGORITHM TO B-52E FLUTTER CONTROL PROBLEM

Introduction

In Section IV, the Moore algorithm was applied to a hypothetical, low order, all pole system. In this section the Moore algorithm is applied to the B-52E flutter control problem, which is a relatively large dimension (24th order), lightly damped, real world system. The airplane is assumed to be in constant altitude, wings-level, steady rectilinear flight (Ref 31).

The attempt here is to compare the input-output properties of the reduced order models obtained via the Moore algorithm with the reduced order models obtained via the AFFDL modified Schwendler and MacNeal technique (Ref 32). The AFFDL technique preserves the physical properties of the original system allowing the resulting reduced order models to be interpreted in a cause-effect (physical relationship) manner. The Moore algorithm, however, preserves only the input-output properties of the original system. Therefore, there will be no attempt to show relationships between the deletion of a state and the physical consequences thereof.

Two measurement devices are available for this test case. They are (see Figure 11): AW 925, which is the vertical wing tip accelerometer, and AW 565, which is the mid-wing vertical accelerometer. Although the test case allows the use of four inputs (elevator, inboard aileron, outboard flaperon, and outboard aileron) (see Figure 12), only the outboard aileron will be utilized for this test case.

This section is organi?ed as follows. First, the acquisition and modelling of the original system is presented. The "impulse balancing" Moore algorithm is then applied to the system. Subsequently, the original system is input to the "step balancing" Moore algorithm. In each case, results are provided and discussed. Finally, the section is summarized and important points are discussed.

Original System Model Acquisition

The B-52E model, upon which this section is based, is discussed in Reference 31. The dynamic equations for the airplane are in a form for use by Level 3.01.02 FLEXSTAB, which is an analysis tool utilized by AFFDL. Unfortunately, the equations are not in state space form required by the Moore algorithm. Therefore, the derivation of the state space form of the original dynamic equations follows.

Let,

$\underline{V}_p$ = $([U \; W \; q]^T)_{3x1}$ (airplane rigid body motion degrees of freedom in the body axis system)

$\underline{r}_{op}$ = $[\; \theta \;]_{1x1}$ (airplane pitch angle)

$\underline{U}_1$ = $([U_{E1} \cdot \cdot \cdot U_{E10}]^T)_{10x1}$ (invacuum structural modes)

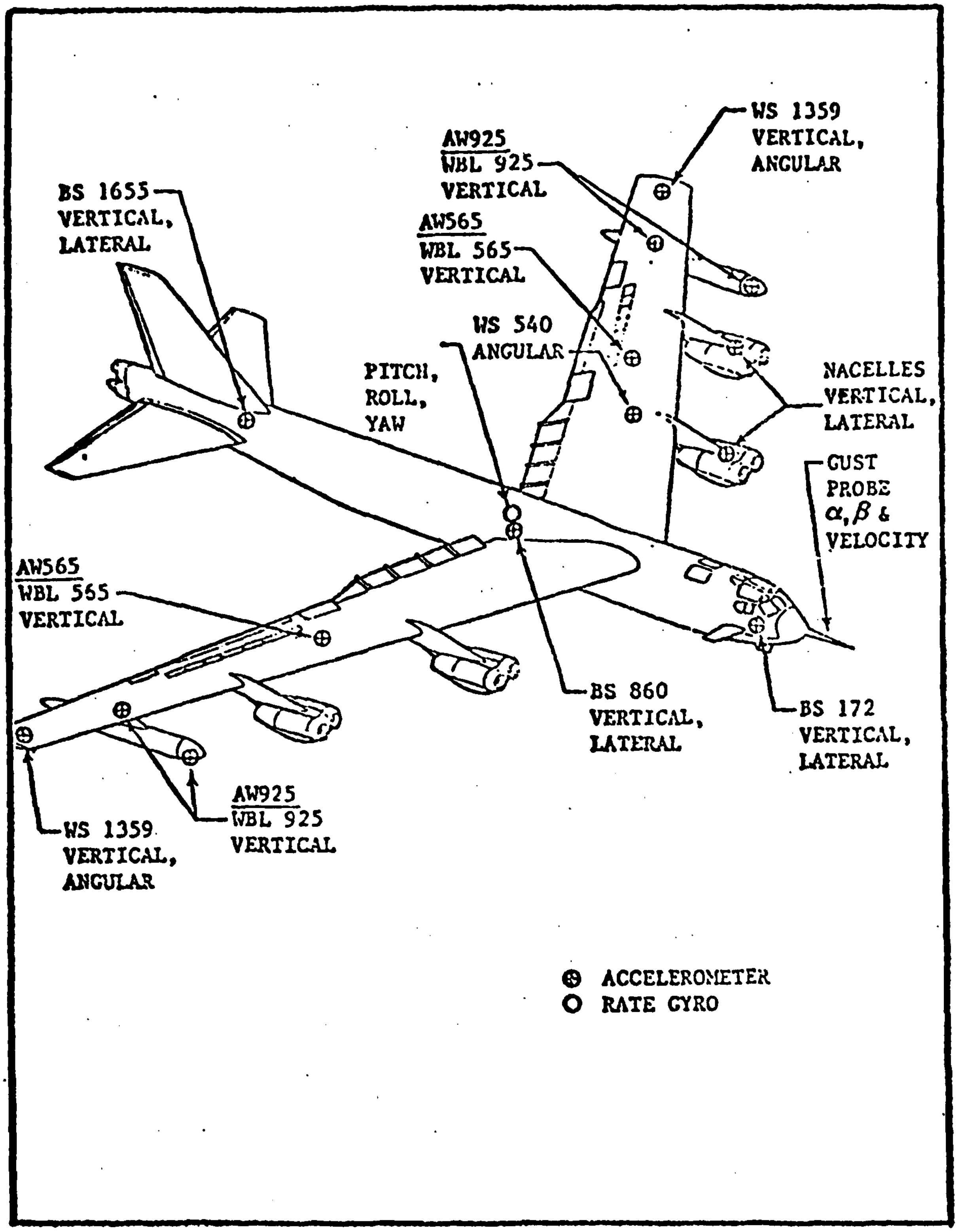


FIGURE 11. Sensor Locations (Ref. 31)

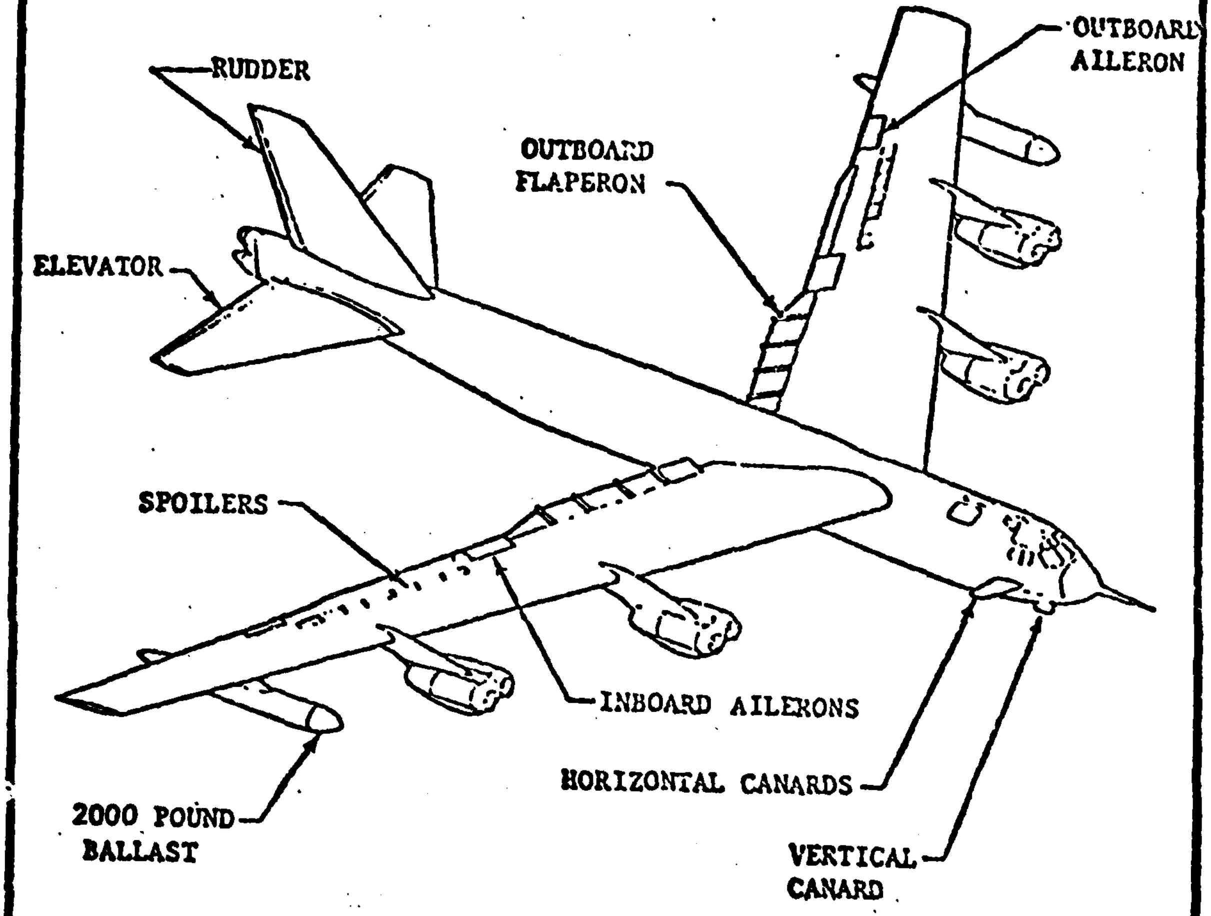


FIGURE 12. B-52E CCV Flight Control Surfaces (Ref. 31)

$\underline{V}_F = \dot{\underline{U}}_1$ where $\dot{\underline{U}}_1 = I\,\underline{V}_F$

$\underline{T}_{2xl}$ = vector of measurements

δc_{4xl} = vector of control inputs

The following equations are the dynamic equations in FLEXSTAB form (ingoring wind gusts) (Ref 31):

$$\dot{\underline{V}}_p = [VP/VP0]\,\underline{V}_p + [VP/UE0]\,\underline{U}_1 + (VP/UE1)\,\dot{\underline{U}}_1 + [VP/DELSO]\,\underline{\delta} c + [VP/R0]\,\underline{r}_{op} \qquad (66)$$

$$\ddot{\underline{U}}_1 = [UE/VP0]\,\underline{V}_p + [UE/UE0]\,\underline{U}_1 + [UE/UE1]\,\dot{\underline{U}}_1 + [UE/DELSO]\,\underline{\delta c} \qquad (67)$$

$$\dot{\underline{r}}_p = [RP/VP0]\,\underline{V}_p + [RP/RP]\,\underline{r}_{op}$$

The measurement equation is:

$$\underline{T} = [T/VP0]\,\underline{V}_p + [T/UE0]\,\underline{U}_1 + [T/UE1]\,\dot{\underline{U}}_1 + [T/UE2]\,\ddot{\underline{U}}_1 + [T/DELSO]\,\underline{\delta}_c + [T/VP1]\,\dot{\underline{V}}_p + [T/R0]\,\underline{r}_p + [T/R1]\,\dot{\underline{r}}_{op} \qquad (68)$$

The measurement equation (Equation 68) has a second derivative term in it. Therefore, by letting $\dot{\underline{U}}_1 = I\,\underline{V}_f$,

$$\ddot{\underline{U}}_1 = I\,\dot{\underline{V}}_f \qquad (69)$$

and augmenting this extra state onto the state vector the problem of the second derivative is reduced to two first order derivatives. The actual state space form of the equations is now presented.

In the state space form, $\dot{\underline{x}} = A\underline{x} + B\underline{u}$, the dynamic equations become

$$\underbrace{\begin{Bmatrix} \dot{\underline{V}}_p \\ \dot{\underline{r}}_{op} \\ \dot{\underline{V}}_f \\ \dot{\underline{U}}_1 \end{Bmatrix}}_{24x1} = \underbrace{\begin{bmatrix} V_p/V_{po} & V_p/R_o & V_p/U_{E1} & V_p/U_{E0} \\ R_p/V_{po} & R_p/R_o & 0 & 0 \\ U_E/V_{po} & 0 & U_E/U_{E1} & U_E/U_{E0} \\ 0 & 0 & I & 0 \end{bmatrix}}_{24x24} \underbrace{\begin{Bmatrix} \underline{V}_p \\ \underline{r}_{op} \\ \underline{V}_f \\ \underline{U}_1 \end{Bmatrix}}_{24x1} + \underbrace{\begin{bmatrix} V_p/DELSO \\ 0 \\ U_E/DELSO \\ 0 \end{bmatrix}}_{24x4} \delta_{c_{4x1}} \tag{70}$$

The measurement equation is:

$$[\underline{T}]_{2x1} = [T/V_{p0} \;\; T/R0 \;\; T/UE1 \;\; T/UE0]_{2x24} \begin{bmatrix} \underline{V}_p \\ \underline{r}_{op} \\ \underline{V}_f \\ \underline{U}_1 \end{bmatrix} + \tag{71}$$

$$+ [T/V_{p1} \;\; T/R_{p1} \;\; T/UE2 \;\; 0]_{2x24} \begin{bmatrix} \dot{\underline{V}}_p \\ \dot{\underline{r}}_{op} \\ \dot{\underline{V}}_f \\ \dot{\underline{U}}_1 \end{bmatrix} + [T/DELSO]_{2x24} \underline{\delta}_c$$

Substituting in for the first order term in equation (71), yields the following measurement equation:

FIGURE 5. Unit Impulse Responses for SISO Example

$$[\underline{T}] = ([T/V_{p0}\ \ T/R_0\ \ T/U_{E1}\ \ T/U_{E0}] + [T/V_{p1}\ \ T/R_{p1}\ \ T/U_{E2}\ \ 0])\,\underline{x}$$
$$+ ([T/V_{p1}\ \ T/R_{p1}\ \ T/U_{E2}\ \ 0]\,[B] + [T/DELSO])\underline{\delta}_c \qquad (72)$$

A computer program was written that formed the A,B,C, and D matrices. Using data supplied by AFFDL, the system was obtained. The A matrix eigenvalues are shown in Table IV.

TABLE IV

The Eigenvalues for the B-52 Test Case

Real Part	± Imaginary Part	Period (sec)
-.3971	.3587E+02	.1752
-.5124	.3603E+02	.1745
-.4651	.2740E+02	.2294
-.1918E+01	.2421E+02	.2604
-.6338	.1996E+02	.3150
-.1445E+01	.1923E+01	.4956
-.9345E-03	.6578E-01	.9556E+02
-.6173	.6154E+01	.1027E+01
-.2137E+01	.1607E+02	.3947
-.05404	.1551E+02	.4051
-.1142	.1238E+02	.5077
-.1464	.1254E+02	.5012

FIGURE 6. Impulse Responses for SISO Example

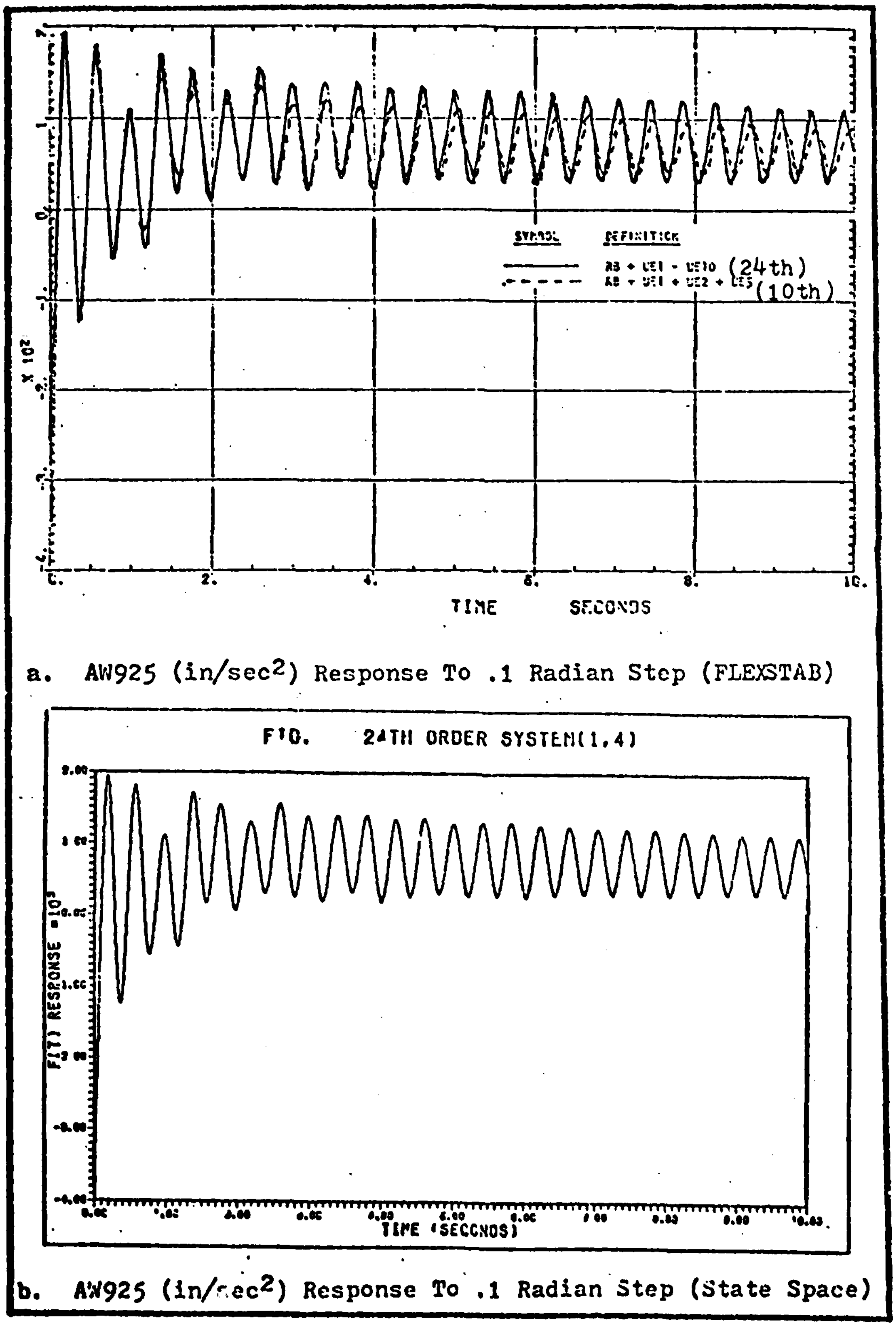
A computer program was written to interface with TOTAL (Ref 21) to obtain time domain and frequency domain responses. The time responses of the full twenty-fourth order model to a .1 radian step input were obtained. Figure 13 shows the comparison of AW925 (in/sec^2) for AFFDL's FLEXSTAB case, and the state space model in response to .1 radian step input. Figure 14 shows the comparison for AW565 (in/sec^2). This verifies that the correct model has been attained. The Moore algorithm is now applied to this state space system.

Application of Moore "Impulse Balancing" Algorithm

The Moore "impulse balancing" algorithm is now applied to the B-52E flutter control problem state space model. To assess the performance of the resulting reduced order model, both the step time response and the frequency response will be compared to the full 24th order system "truth model".

Because we examine the time response performance with a step input an important point should be ncted. In Section II, theory was presented that claimed if the algorithm is designed for the particular input that the reduced order models will be evaluated against, then lower order models can probably be attained for similar levels of reproduction accuracy. By subjecting the B-52E flutter control problem reduced order models to a step input, for either the impulse balanced reduced

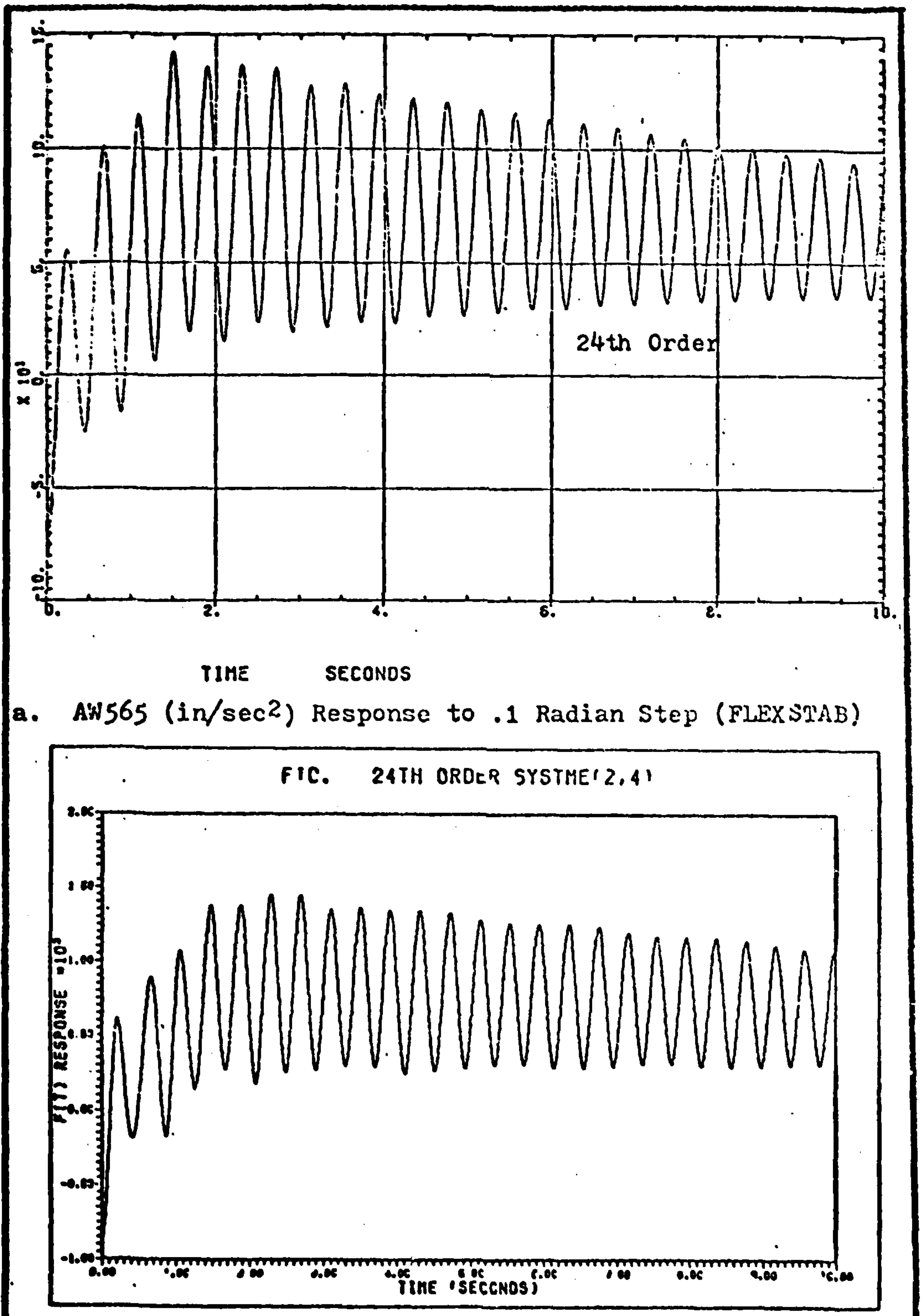
FIGURE 7. Magnitude Responses for SISO Example



a. AW925 (in/sec^2) Response To .1 Radian Step (FLEXSTAB)

b. AW925 (in/sec^2) Response To .1 Radian Step (State Space)

FIGURE 13. Response To .1 Radian Step in Outboard Aileron



a. AW565 (in/sec^2) Response to .1 Radian Step (FLEXSTAB)

b. AW565 (in/sec^2) Response To .1 Radian Step (STATE SPACE)

FIGURE 14. Response To .1 Radian Step in Outboard Aileron

FIGURE 8. Magnitude Responses for SISO Example

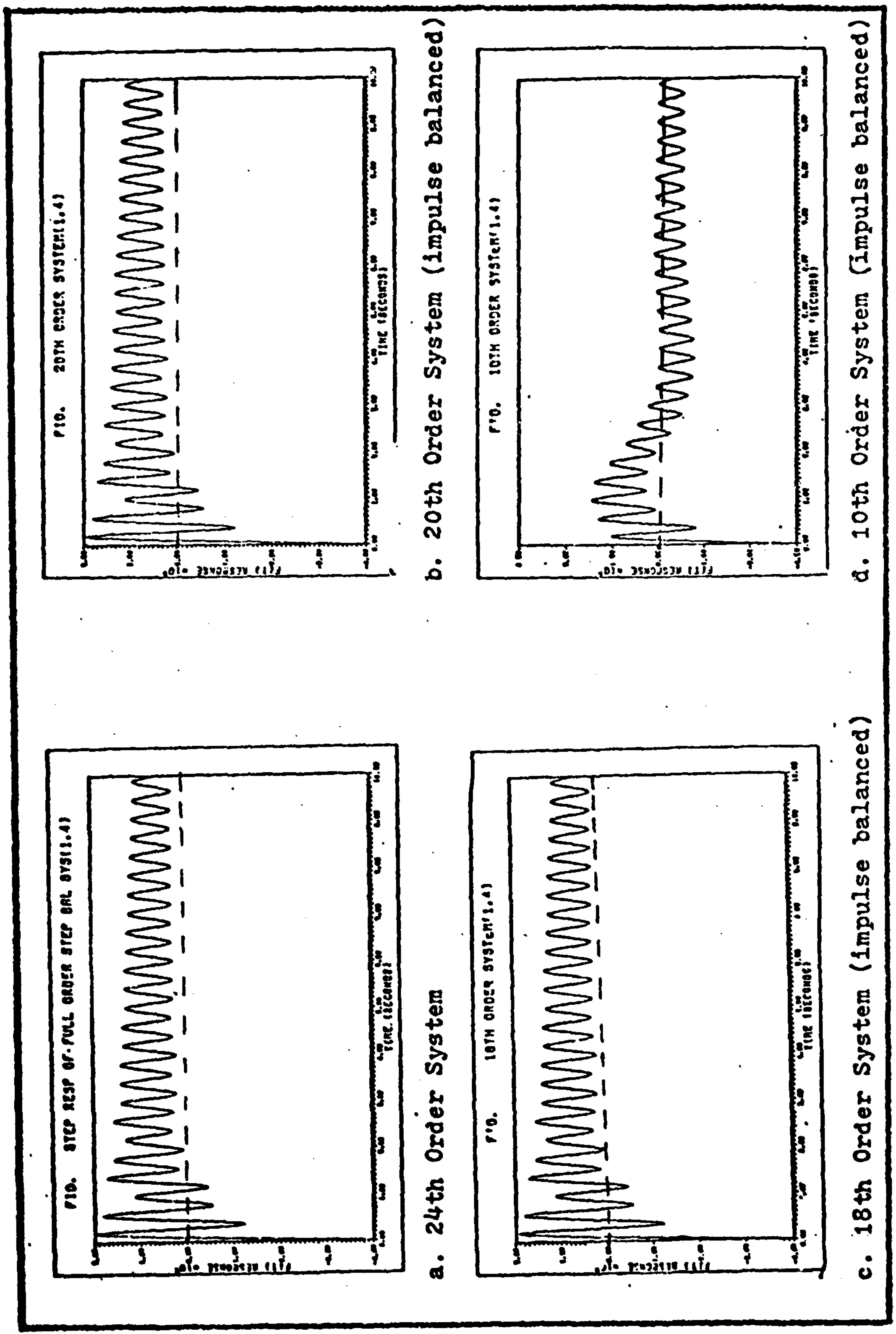
order model, or the step balanced reduced order model, we would expect that the step balancing algorithm should give better time response performance results.

Results. Figure 15 contains the response of AW925 (vertical wing-tip accelerometer) to a .1 radian step in outboard aileron for the full order original system, and the twentieth, eighteenth, and tenth order impulse balanced reduced order models respectively. A definite degradation in both transient response and steady state response occurs as the system order is reduced. Similar results are pictured in Figure 16 for AW565 (vertical mid-wing accelerometer) for a .1 radian step in outboard aileron.

In Figure 17, the frequency response for AW925/OBA is shown for the twenty-fourth (full order), twentieth, eighteenth, and tenth order systems. As the system order is reduced, essential high frequency information is being lost. However, the low frequency portion of the responses seems to be relatively constant. Figure 18 shows similar results for AW565/OBA.

Figure 19 and 20 show the phase responses of AW925/OBA and AW565/OBA, respectively. As in Figures 17 and 18 the high frequency information is being discarded. This suggests that sinusoidal inputs at these higher frequencies to the reduced order systems would not produce accurate replication of the full-order system.

FIGURE 9. Phase Responses for SISO Example



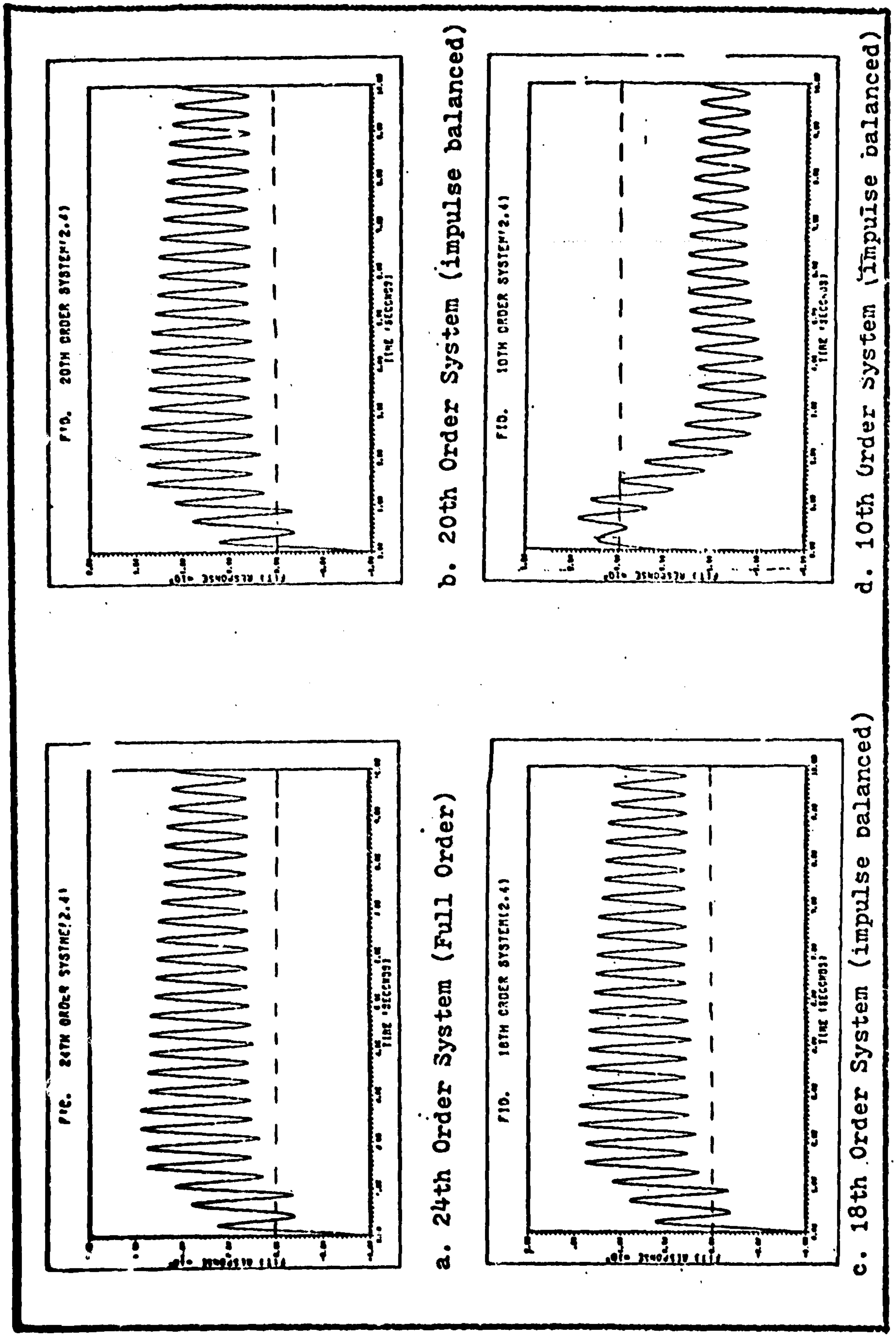
a. 24th Order System

b. 20th Order System (impulse balanced)

c. 18th Order System (impulse balanced)

d. 10th Order System (impulse balanced)

FIGURE 15. AW925/OBA To .1 Rad. Step Input Using Impulse Balancing



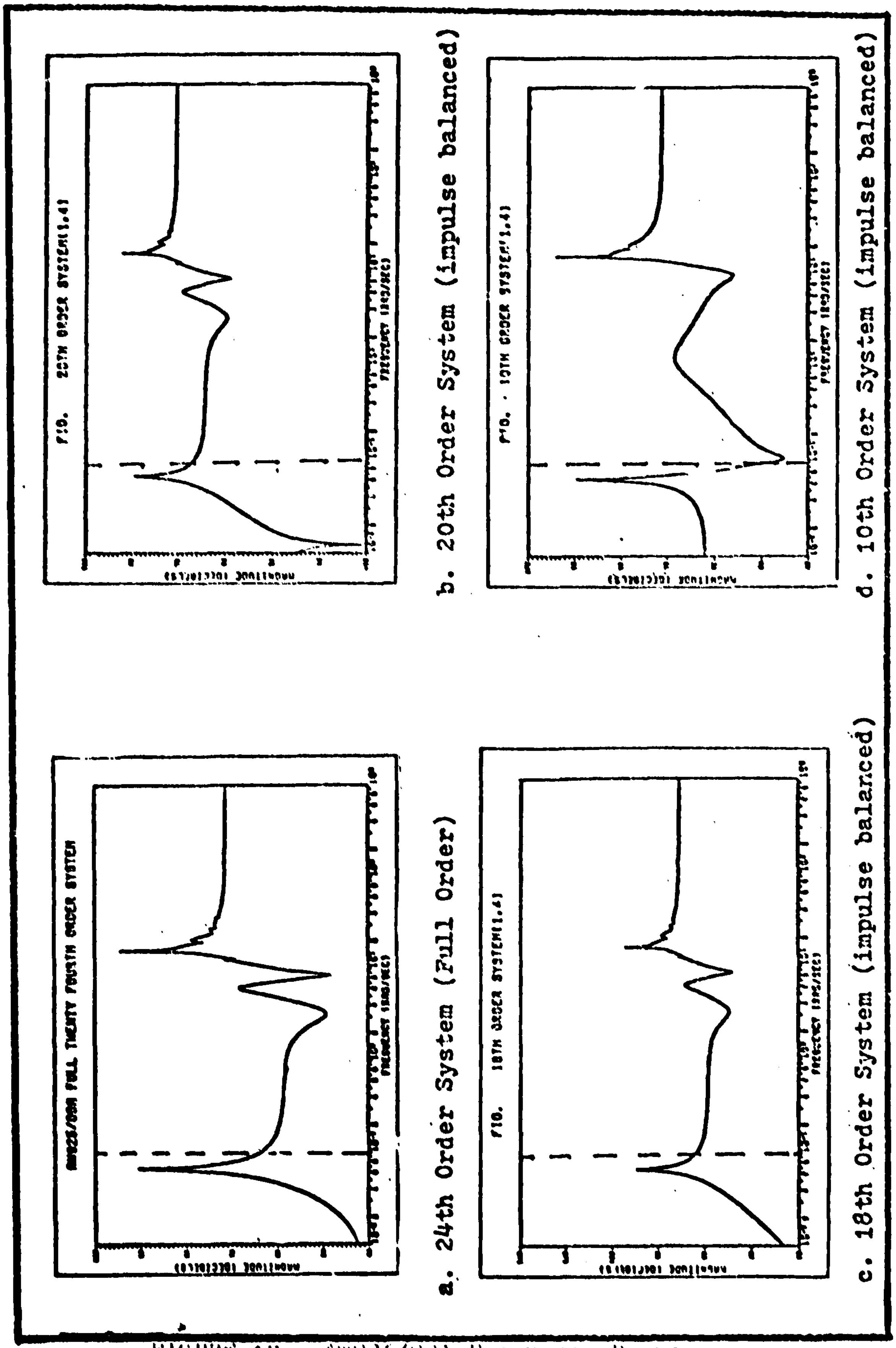
a. 24th Order System (Full Order)

b. 20th Order System (impulse balanced)

c. 18th Order System (impulse balanced)

d. 10th Order System (impulse balanced)

FIGURE 16. AW565/OBA To .1 Rad. Step Input
Using Impulse balancing



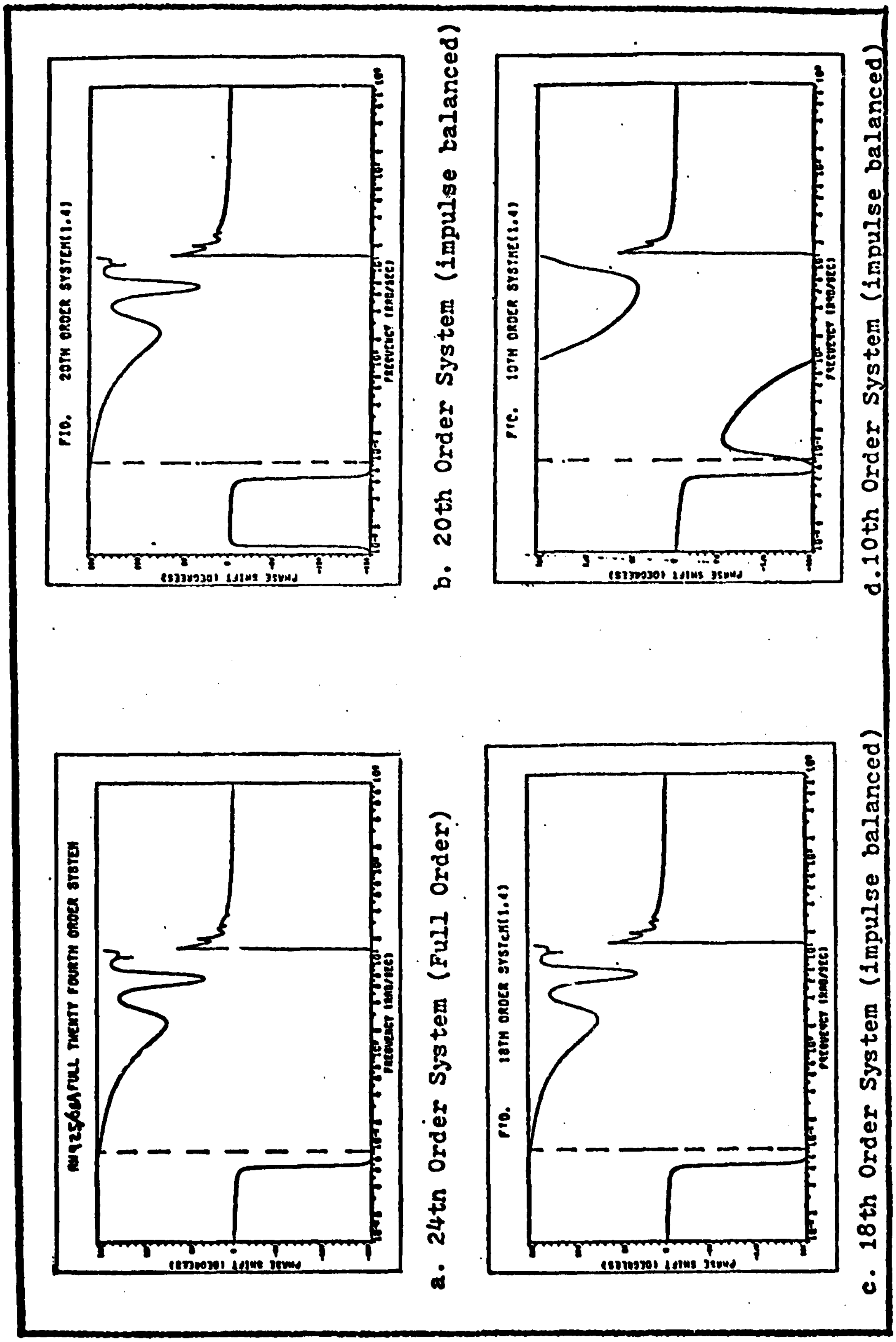
a. 24th Order System (Full Order)

b. 20th Order System (impulse balanced)

c. 18th Order System (impulse balanced)

d. 10th Order System (impulse balanced)

FIGURE 17. AW925/OBA Frequency Response Using Impulse Balancing



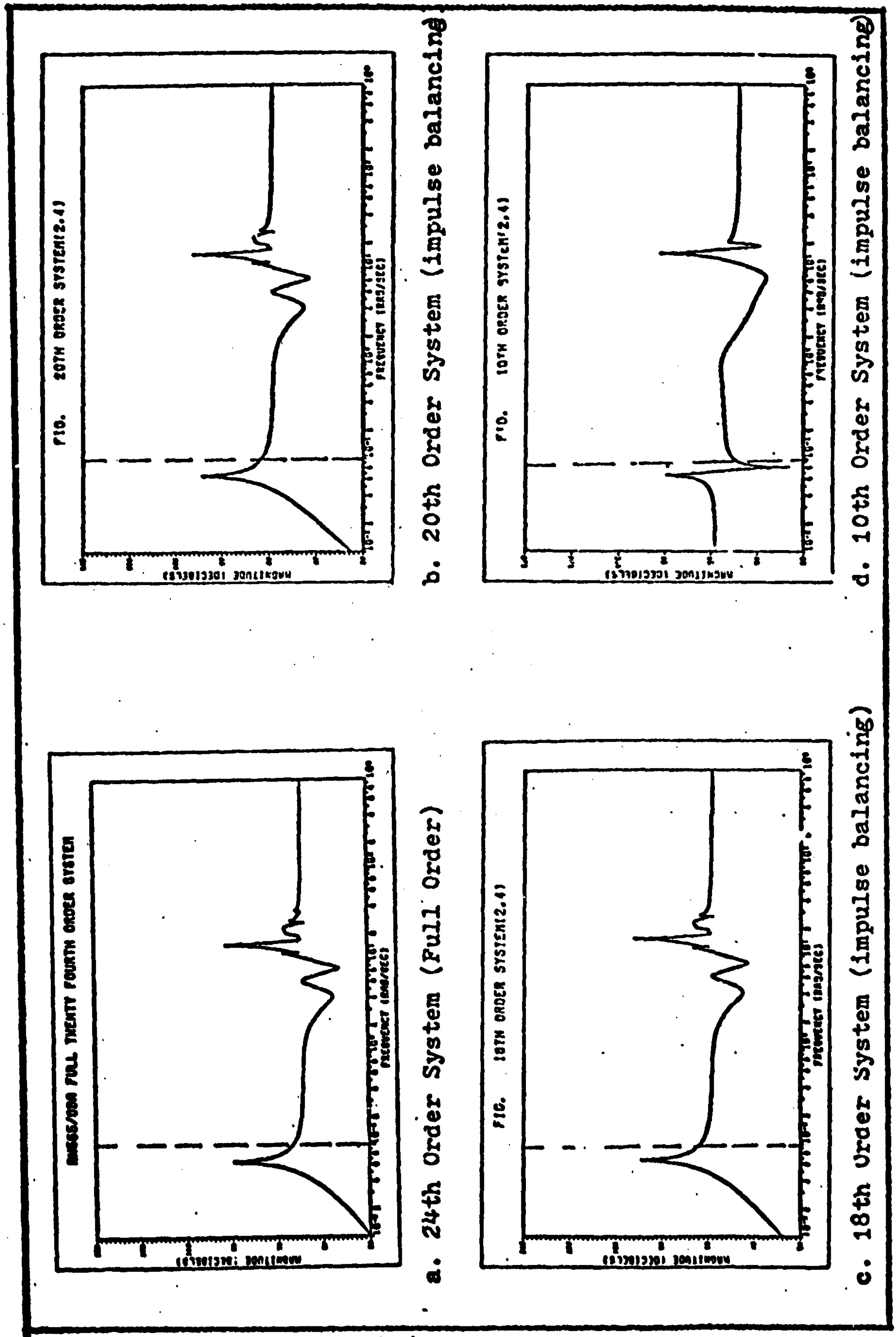
a. 24tn Order System (Full Order)

b. 20th Order System (impulse balanced)

c. 18th Order System (impulse balanced)

d.10th Order System (impulse balanced)

FIGURE 18. AW925/OBA Phase Response
Using Impulse Balancing



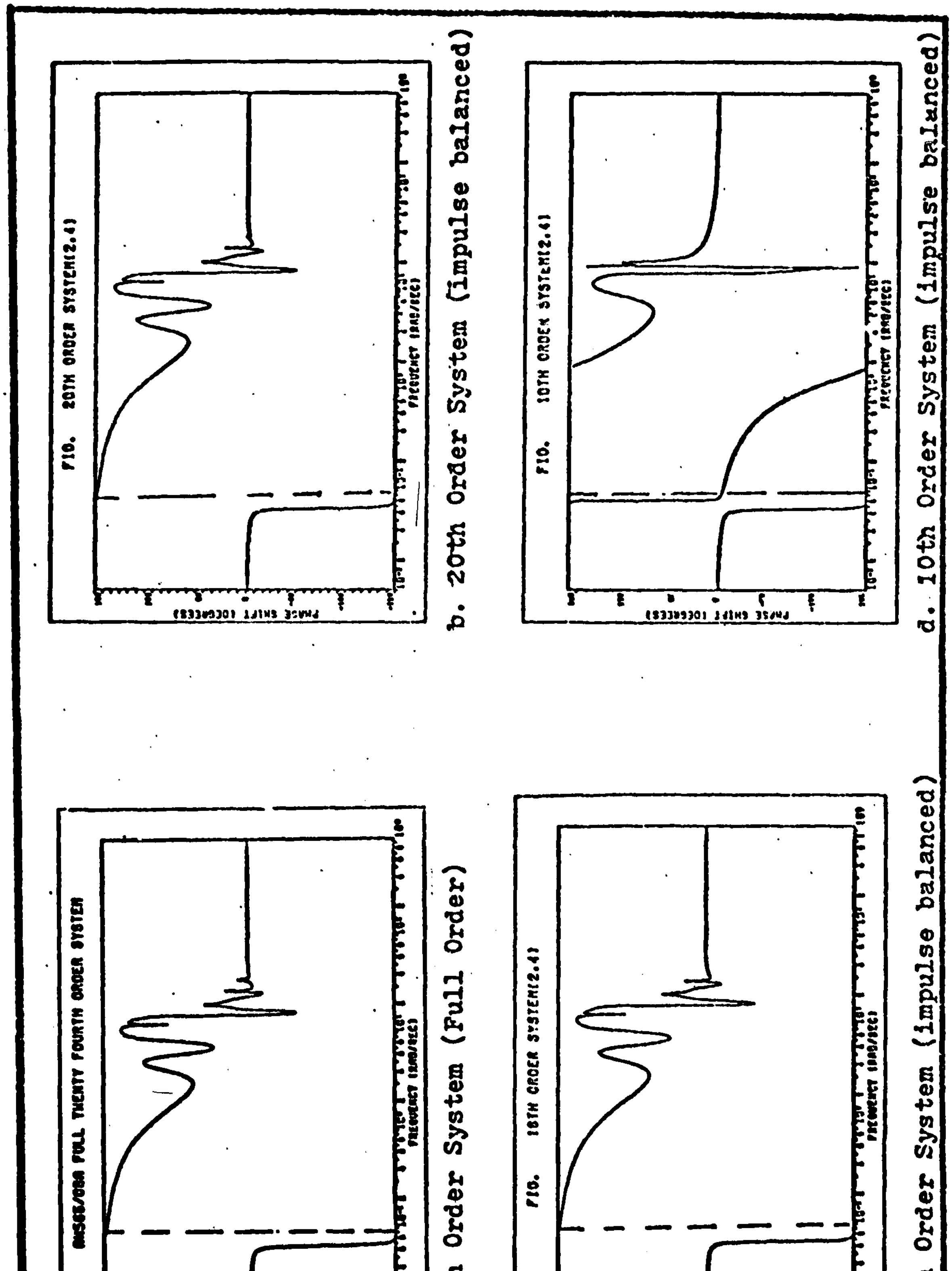
a. 24th Order System (Full Order)

b. 20th Order System (impulse balancing

c. 18th Order System (impulse balancing)

d. 10th Order System (impulse balancing)

FIGURE 19. AW565/OBA Frequency Response Using Impulse Balancing



a. 24th Order System (Full Order)

b. 20th Order System (impulse balanced)

c. 18th Order System (impulse balanced)

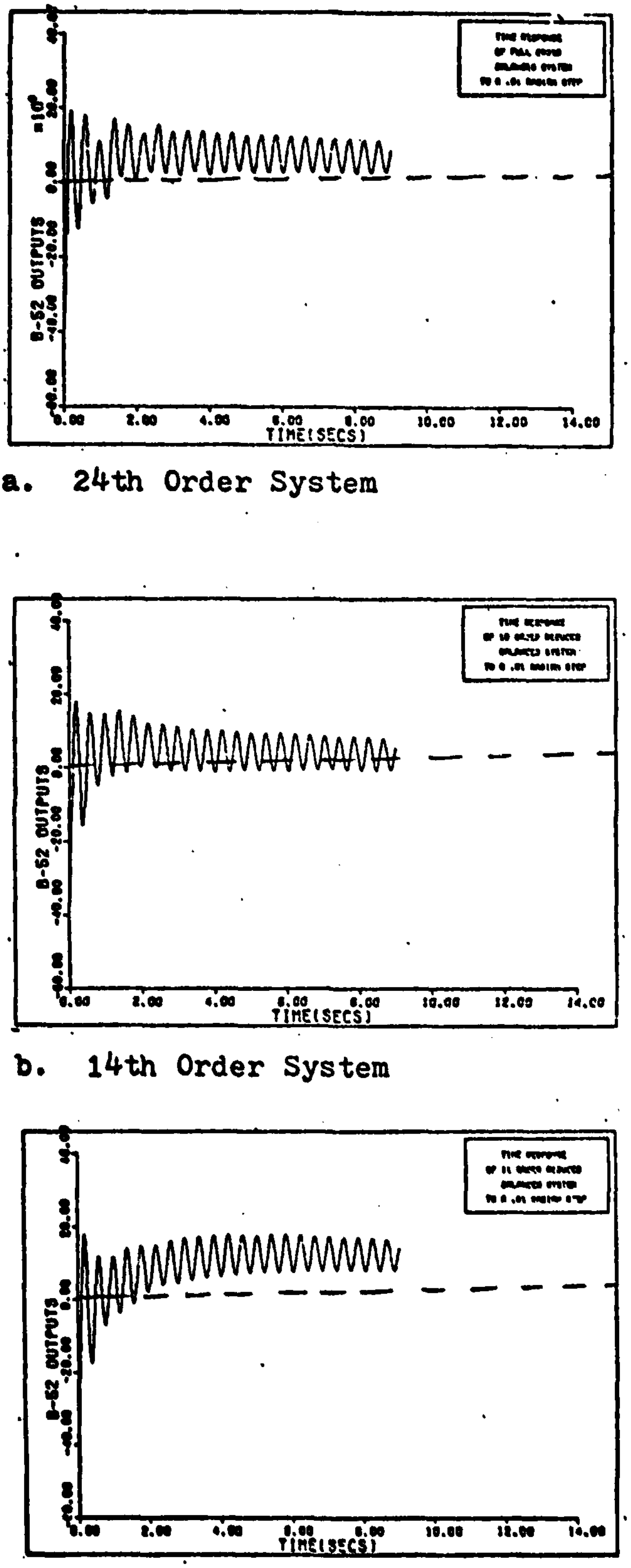
d. 10th Order System (impulse balanced)

FIGURE 20. AW565/OBA Phase Response
Using Impulse Balancing

Figures 21 and 22 show the responses of AW925 and AW565 to a .1 radian step in outboard aileron respectively. These plots show the full twenty-fourth order, the fourteenth order, and the thirteenth order systems for both Figures 21 and 22. There is a slight steady state offset occurring in 21b and 22b (fourteenth order). However, severe transient and steady state errors exist in the thirteenth order models in Figures 21c and 22c. This suggests that fourteenth order is the lowest order this system can be reduced to when impulse balancing is applied but a step is the actual input used for evaluation.

H_{INF} Matrix Investigation. As mentioned in Section II, Appendix B, and References 24,25, there is a byproduct of this algorithm which gives an estimate of the lowest possible order the system can be reduced to without severe loss of reproduction accuracy (subjective). This byproduct is referred to by Moore in References 24 and 25 as the "H_{INF}" matrix.

In theory, the singular values of the H_{INF} matrix provide a guideline to the choice of the lowest possible order the system can be reduced for accurate original system replication. However, this test case uses a step input to an impulse balanced system. Therefore, the results may not be as accurate as later in this section when the system is subjected to step balancing.

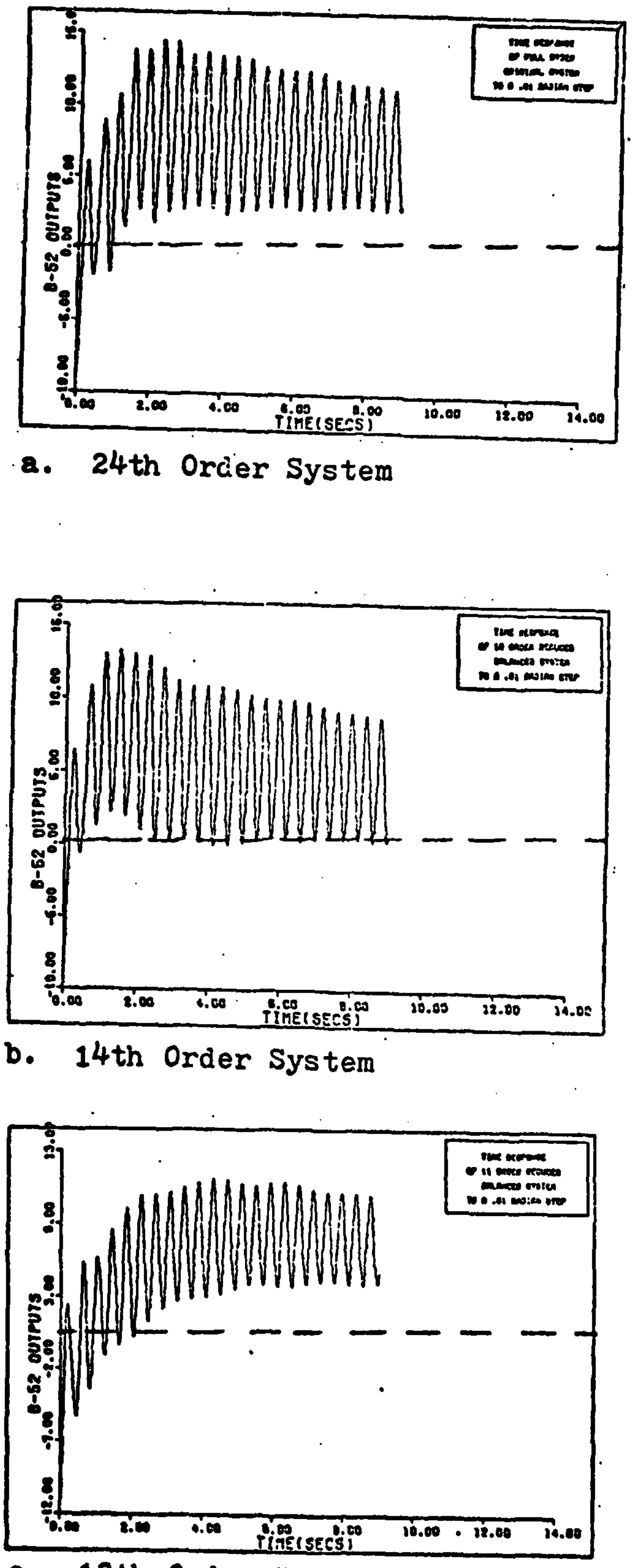


a. 24th Order System

b. 14th Order System

c. 13th Order System

FIGURE 21. AW925/OBA to .1 Radian Step Input Using Impulse Balancing



a. 24th Order System

b. 14th Order System

c. 13th Order System

FIGURE 22. AW565/OBA to .1 Radian Step Input Using Impulse Balancing

The singular values of the H_{INF} matrix for impulse balancing are shown in Table V. The singular values are listed in descending order. As discussed in Section II the number of "large" singular values that cluster together provide a guideline for the choice of minimum dimension of the reduced order model. There is no clear dividing line. However, one of the largest ratios is between the fifth and sixth singular values. This indicates that for impulse balancing coupled with an impulse input, the system order could possibly be reduced to fifth order without severe loss of accuracy. Unfortunately, this system has a D matrix (feedforward) and an impulse input cannot be applied.

TABLE V.

Singular Values of the H_{INF} Matrix for Impulse Balancing

2.2443E+05	8.8185E+03	1.8684E+03	3.3900E+02
2.1825E+05	6.0901E+03	2.1874E+03	3.3819E+02
1.3755E+05	5.2009E+03	2.1445E+03	2.9344E+02
1.3723E+05	3.6827E+03	2.0217E+03	2.8774E+02
1.2722E+05	3.5674E+03	1.4901E+03	9.3683E+01
1.2131E+04	3.1964E+03	1.3496E+03	9.2485E+01

The fourteenth order impulse balanced system produces a relatively good reproduction of the original system in response to a .1 radian step in outboard aileron. However, if the theory holds, a definite improvement (lower attainable system order) should be achieved through the use of step balancing.

Application of Moore "Step Balancing" Algorithm

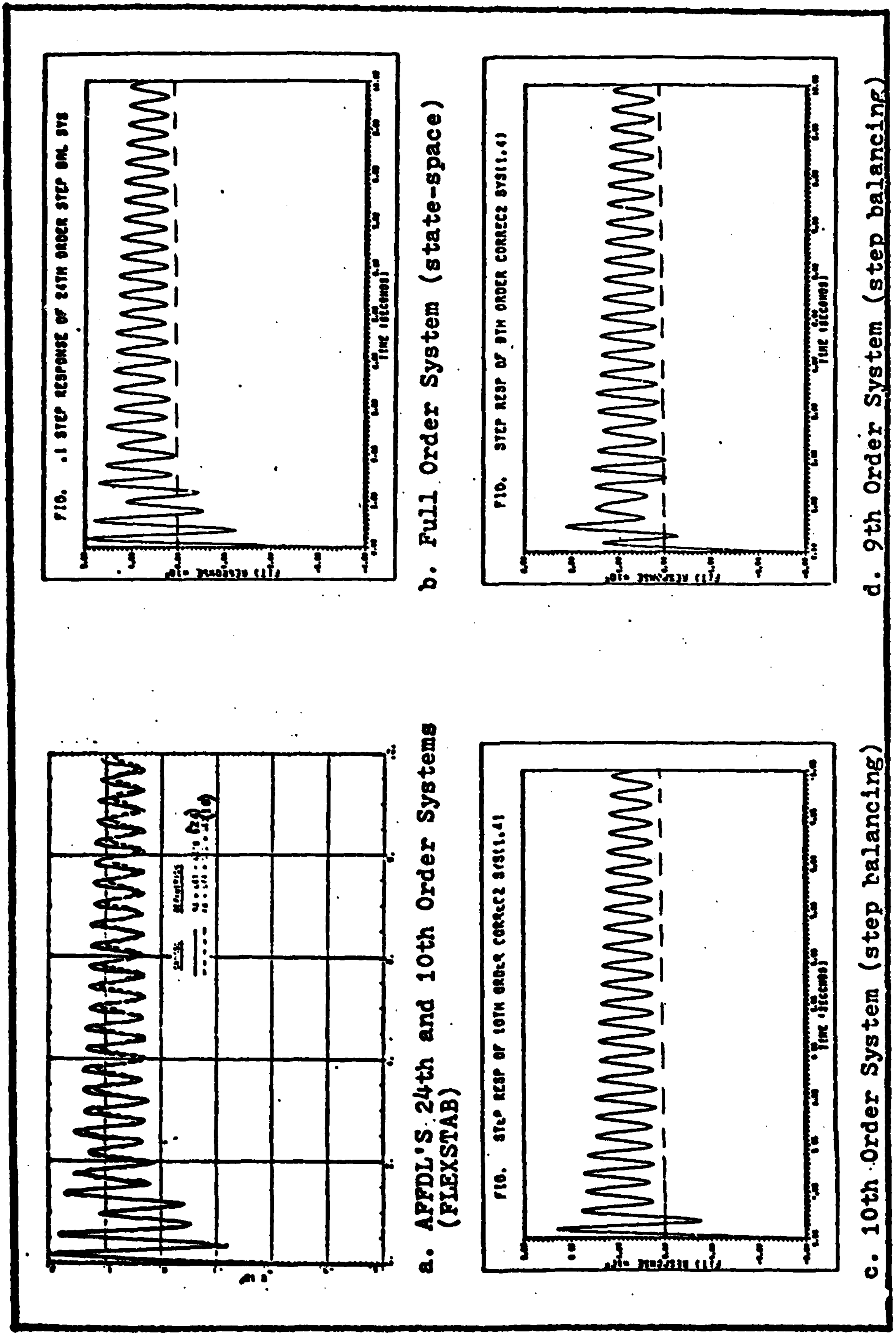
By utilizing the two modifications to the impulse balancing algorithm described in Section II, specifically:

1) input $(A, A^{-1}B, C)$ (provided A^{-1} exists) to the impulse balancing algorithm

2) add a D matrix to compensate for steady state offset

the step balancing algorithm is realized. The results will show that when the algorithm is "optimized" for a given input, in this case a step input, lower order models are achievable for a given level of system reproduction accuracy.

Results. In Figure 23, AW925 to a .1 radian step in outboard aileron is shown for the full twenty fourth order (AFFDL's and this thesis'), the tenth order (AFFDL's and this thesis'), and the ninth order (this thesis') models. In Figure 23a AFFDL's twenty fourth order and tenth order systems are shown. The solid curve is the twenty fourth order system; the dashed curve, the tenth order model. Notice that the tenth order response has a definite phase shift as time increases. It also seems to be damped slightly more than the original system. However, it gives an excellent reproduction of the transient



a. AFFDL'S 24th and 10th Order Systems (FLEXSTAB)

b. Full Order System (state-space)

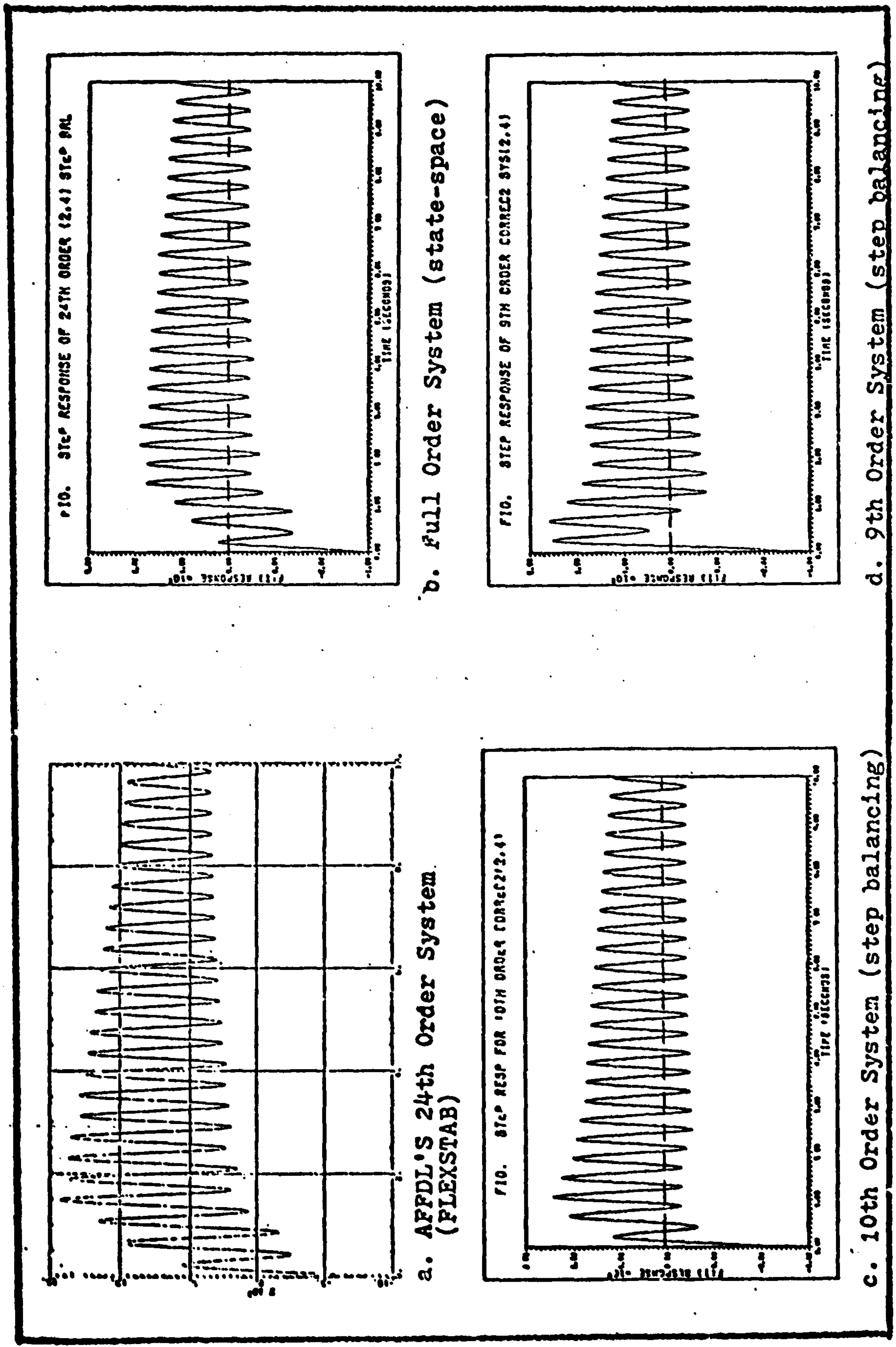
c. 10th Order System (step balancing)

d. 9th Order System (step balancing)

FIGURE 23. AW925/OBA to .1 Rad. Step Input Using Step Balancing

response and the frequency of oscillation. Figures 23c and 23d show the tenth and ninth order models obtained via the Moore step balancing algorithm. The transient reproduction ability is low for both of these systems. However, there is no phase shift or excessive damping as found in the AFFDL model. Figure 24 shows similar results for AW565 to a .1 radian step in outboard aileron. Table VI shows the percent differences in magnitude between the twenty fourth order system and the tenth and ninth order systems for both AW925 and AW565 as a function of time. The table illustrates the large magnitude of transient errors and the small magnitude of steady state errors.

The frequency responses of the twenty fourth, tenth, and ninth order systems for AW925/OBA and AW565/OBA are shown in Figures 25 and 26, respectively. Figures 27 and 28 show the phase responses for AW925/OBA and AW565/OBA, respectively. Important high frequency, magnitude and phase information is being lost as the system order is reduced. However, comparing Figures 25 with 17 for impulse balancing, 26 with 19 for impulse balancing, 27 with 19 for impulse balancing, and 28 with 20, the amount of high frequency information retained by the step balanced reduced order models shows a definite improvement over the impulse balanced reduced order model. Also the tenth order time response for step balancing (Figures 23c, 24c) are



a. AFFDL'S 24th Order System (FLEXSTAB)

b. Full Order System (state-space)

c. 10th Order System (step balancing)

d. 9th Order System (step balancing)

FIGURE 24. AW565/OBA to .1 Rad. Step Input Using Step Balancing

TABLE VI

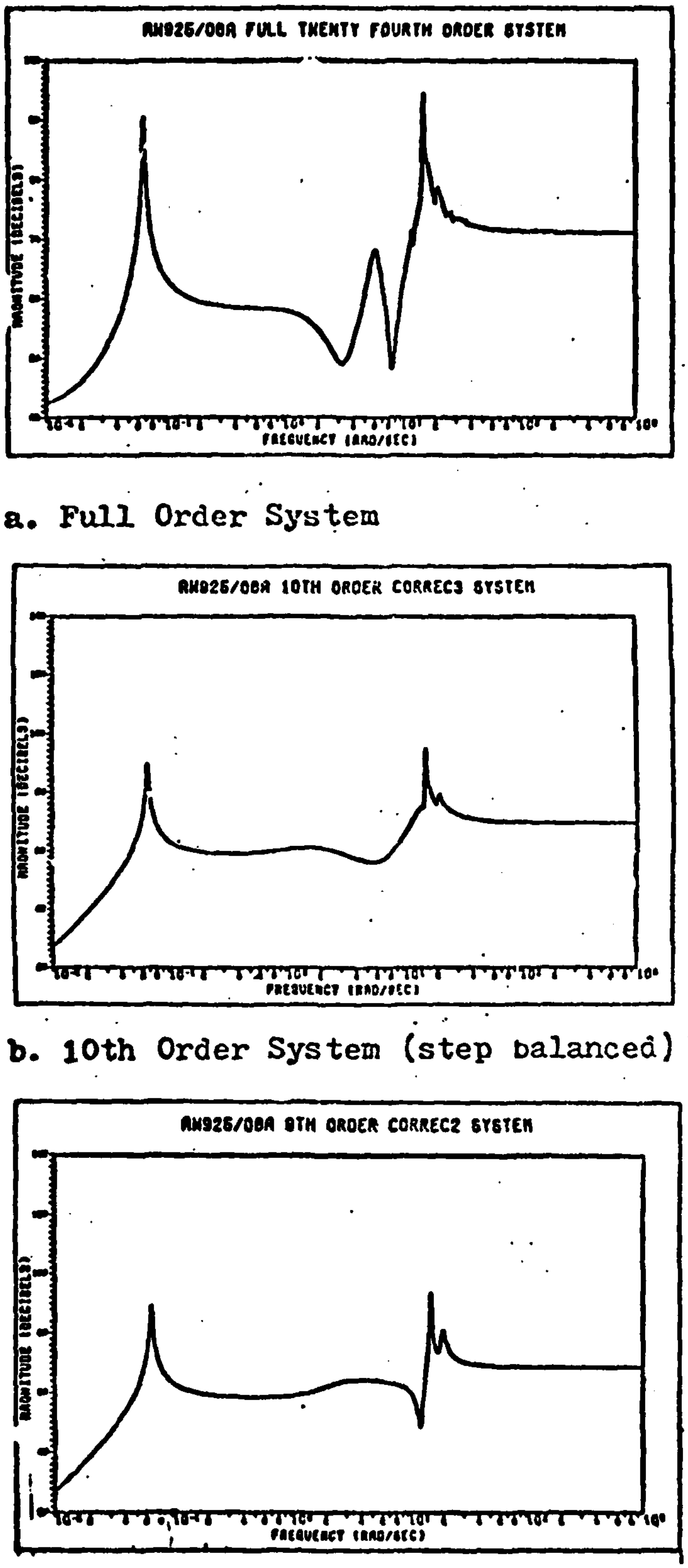
Differences in Time Response Between 24th and 10th and 9th Order Systems

T	DIF. AW925	DIF.	DIF. AW565	DIF.
	24-10(%)	24-9(%)	24-10(%)	24-9(%)
0	0	0	0	0
.1	69.58	318.61	214.51	87.23
.2	16.52	35.34	43.98	58.71
.3	254.58	101.17	64.81	83.7
.4	64.65	123.83	179.32	117.95
.5	2.07	47.66	95.25	97.25
.6	4.11	1.79	50.53	56.62
.7	81.05	73.31	40.98	89.33
.8	157.66	160.33	71.16	79.83
.9	47.9	59.89	123.72	125.37
1.0	35.91	20.16	44.42	35.37
1.2	197.26	218.0	48.48	41.31
1.4	4.48	8.69	23.77	5.43
1.6	1.16	104.76	17.96	15.80
1.8	8.95	7.81	10.22	6.16
2.0	61.34	92.13	15.74	6.04
2.2	8.97	4.49	2.79	10.59
2.4	6.98	29.21	1.71	9.32
2.6	6.76	16.24	3.00	11.633
2.8	26.85	25.77	5.26	6.67
3.0	7.84	7.21	4.64	3.73
4.0	11.02	20.8	3.19	4.92
5.0	5.61	7.28	3.77	5.32
6.0	1.64	6.48	2.42	2.99
7.0	5.04	5.21	4.54	4.41
8.0	2.23	7.52	2.13	2.7
9.0	5.94	3.56	7.2	6.7
10.0	7.17	3.42	4.72	4.06

AD-A080 371 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION! --ETC(U)
DEC 79 J R MCCLENDON
UNCLASSIFIED AFIT/GE/EE/79-22 NL

2 OF 3

AD A08037

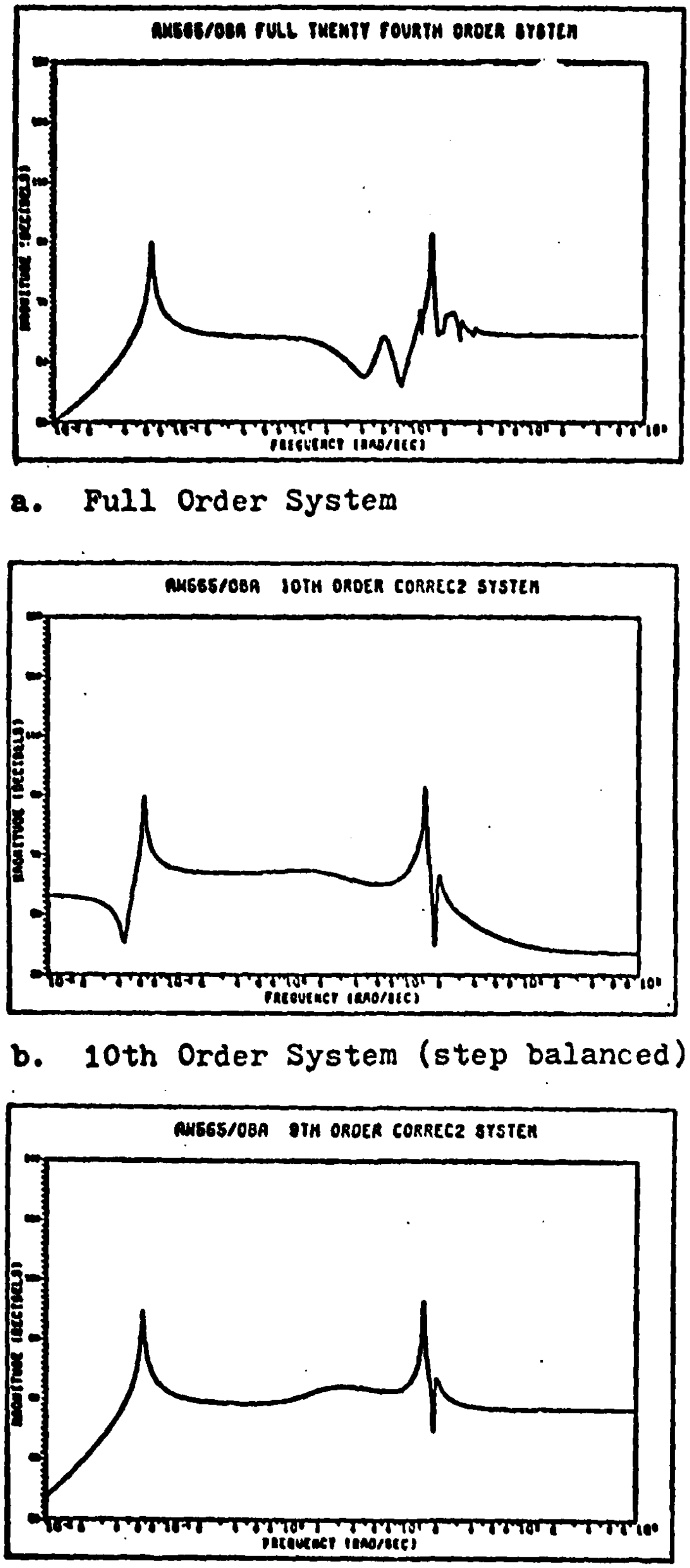


a. Full Order System

b. 10th Order System (step balanced)

c. 9th Order System (step balancing)

FIGURE 25. Frequency Response (AW925/OBA) Using Step Balancing

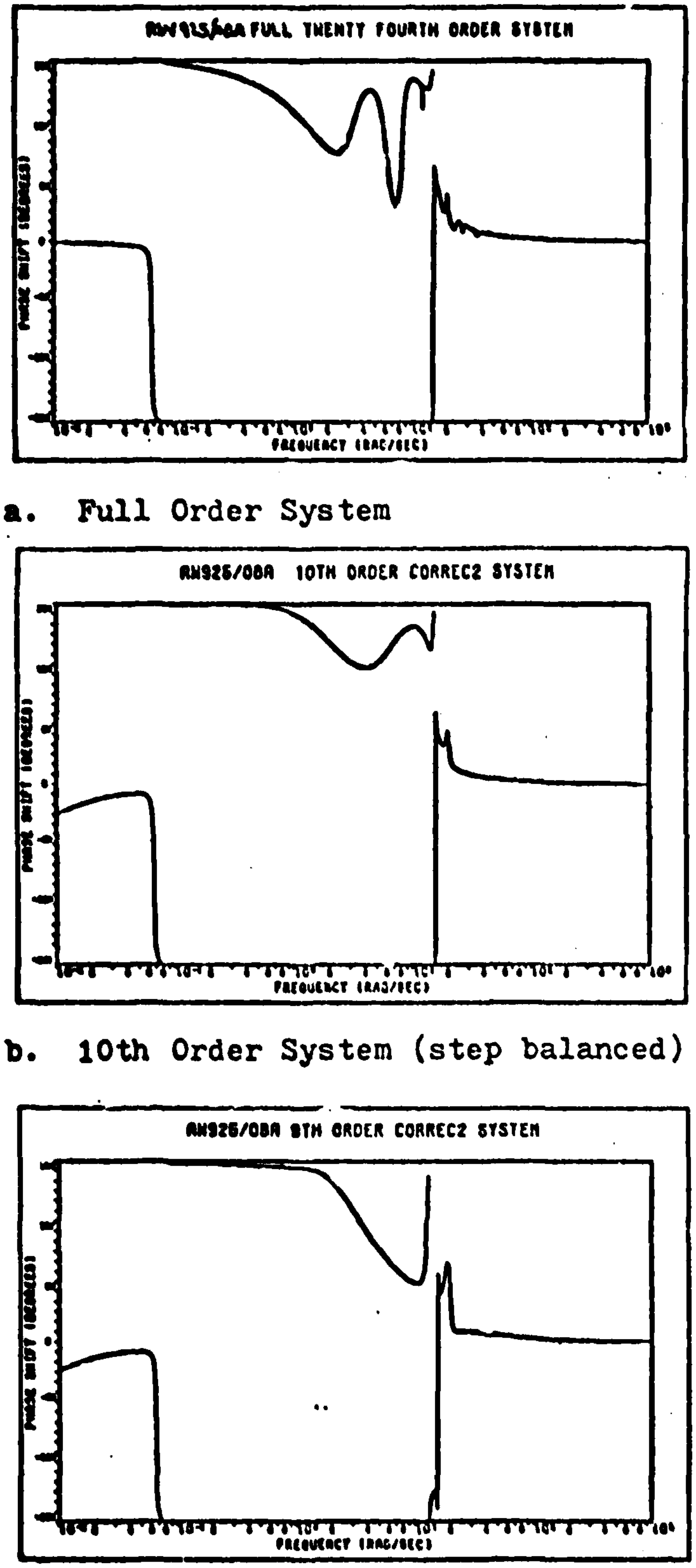


a. Full Order System

b. 10th Order System (step balanced)

c. 9th Order System (step balanced)

FIGURE 26. Frequency Response (AW565/OBA) Using Step Balancing



AW925/OBA FULL TWENTY FOURTH ORDER SYSTEM

a. Full Order System

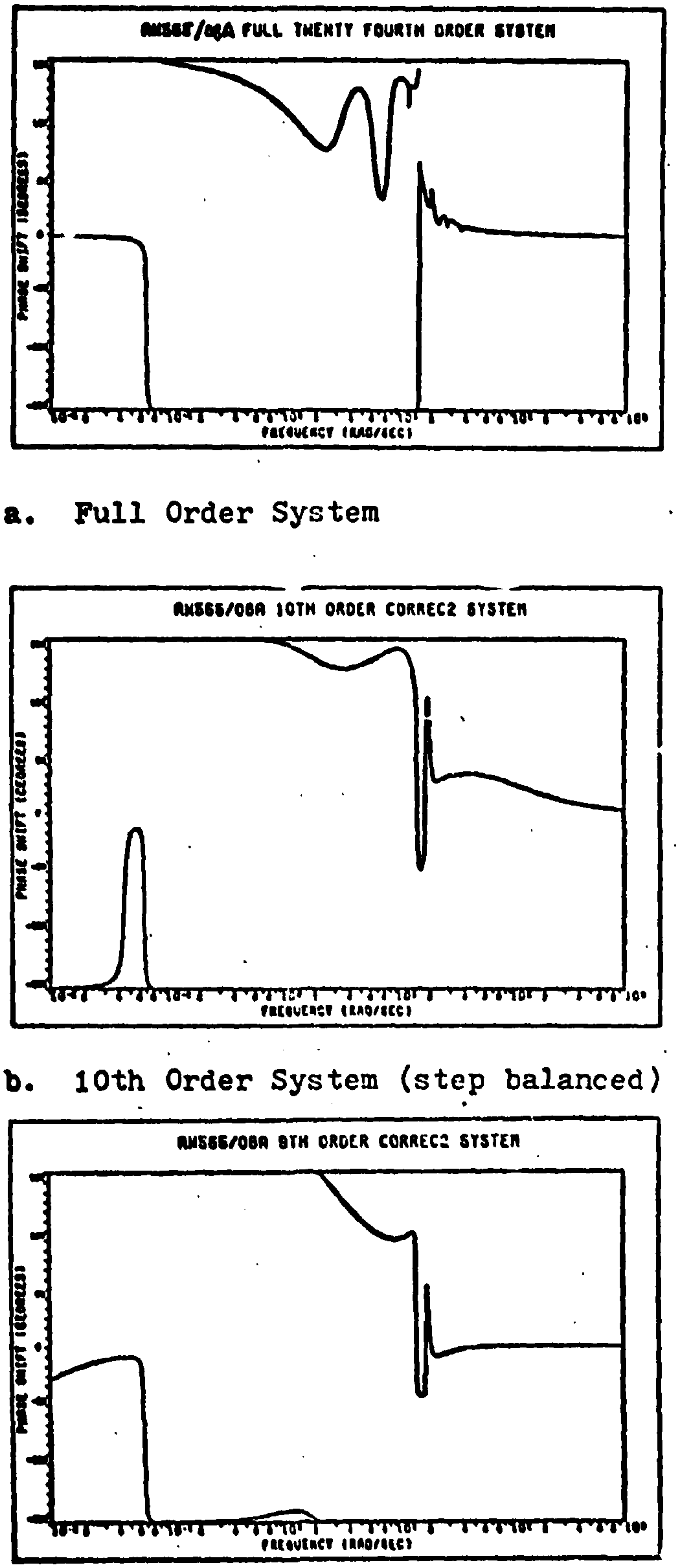
AW925/OBA 10TH ORDER CORREC2 SYSTEM

b. 10th Order System (step balanced)

AW925/OBA 9TH ORDER CORREC2 SYSTEM

c. 9th Order System (step balanced)

FIGURE 27. Phase Response (AW925/OBA)
Using Step Balancing



a. Full Order System

b. 10th Order System (step balanced)

c. 9th Order System (step balanced)

FIGURE 28. Phase Response (AW565/OBA) Using Step Balancing

much more representative of the original system than the tenth order system obtained via impulse balancing (Figures 15d, 16d). In addition to improvement in results by using step balancing, the H_{INF} matrix also gives a better guideline for the choice of the lowest order reduced order model than it did for impulse balancing.

H_{INF} Matrix Investigation. Table VII contains the singular values of the H_{INF} matrix for step balancing. Grouping these singular values into "large" and "small" groups yields approximately two singular values in the "large" group. If adhered to strictly, this suggests second order as being the lowest order attainable without severe loss of reproduction accuracy. But if followed as a guideline, possibly fifth or sixth order is the lowest order achievable (subjective).

TABLE VII

Singular Values of the H_{INF} Matrix for Step Balancing

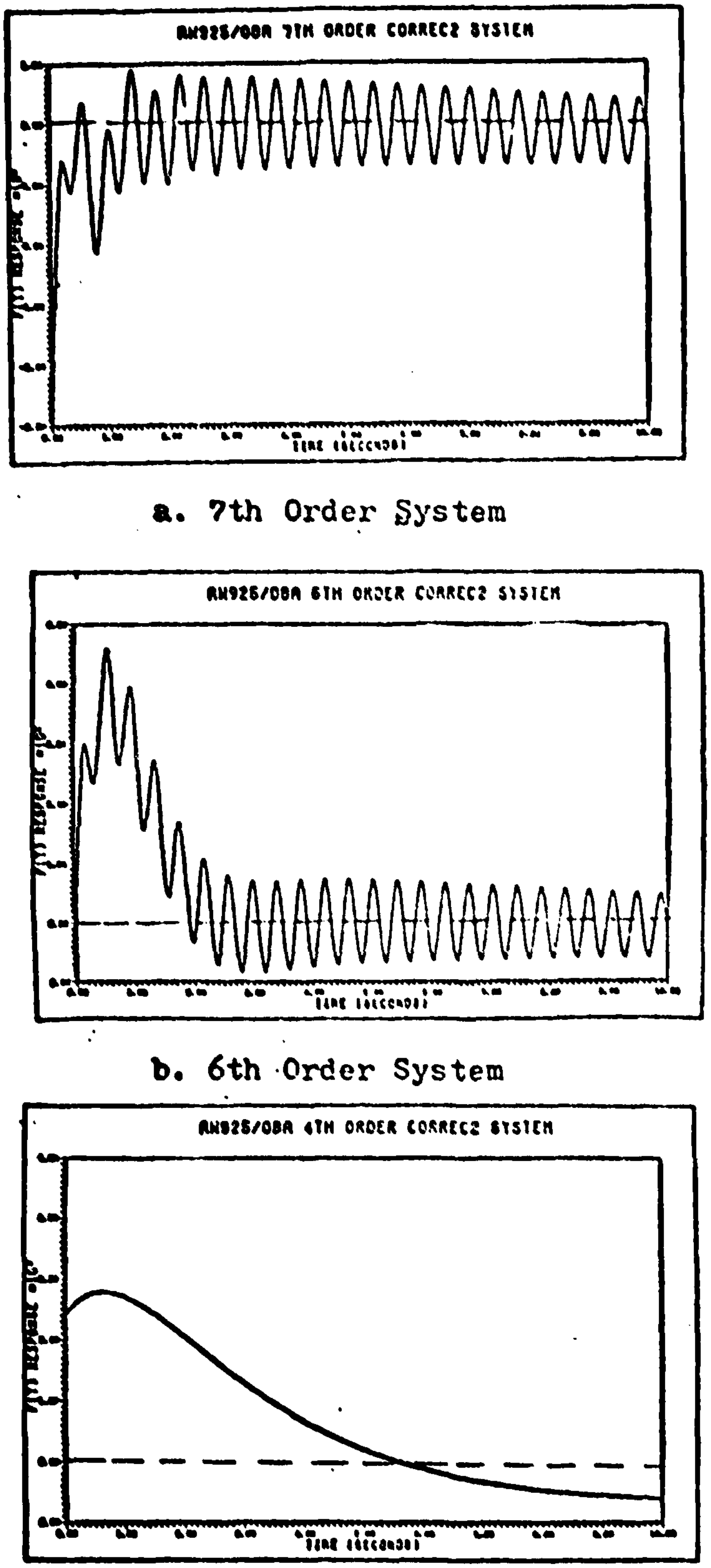
3.3665E+06	6.2795E+02	2.5876E+02	2.7487E+01
3.3625E+06	5.9453E+02	2.3815E+02	2.7123E+01
8.8716E+03	3.3029E+02	1.2885E+02	8.1539E+00
8.8448E+03	3.0580E+02	1.1655E+02	8.0367E+00
6.0047E+03	2.8524E+02	8.1622E+01	2.5982E+00
9.0330E+02	2.7912E+02	8.0676E+01	2.5818E+00

Figure 29 shows the seventh, sixth, and fourth order systems for AW925 in response to a .1 radian step in outboard aileron. Figures 30 and 31 show the frequency and phase responses for AW925/OBA respectively. Highly evident in Figures 30 and 31 is the fact that as the system order is reduced, more and more high frequency information is lost. This is occurring at lower frequencies as the system order is reduced.

Therefore, depending on an observer's idea of "accurate" reproduction of the original system, many different system orders are achievable. However, approximately tenth order is the lowest order system attainable while still reproducing both the transient and steady state portions of the response fairly well.

Summary

The Moore algorithm seems to provide a viable method for obtaining a reduced order model for a given input system. Rather than delete nondominant poles (if any exist), as many algorithms do, the poles and zeros as discussed in Section II are shifted to compensate for the loss of a state. The Moore algorithm in theory is purging the model of the uncontrollable, unobservable portion of the system as system order is reduced. In practice, the Moore algorithm is discarding high frequency information as system order is reduced.

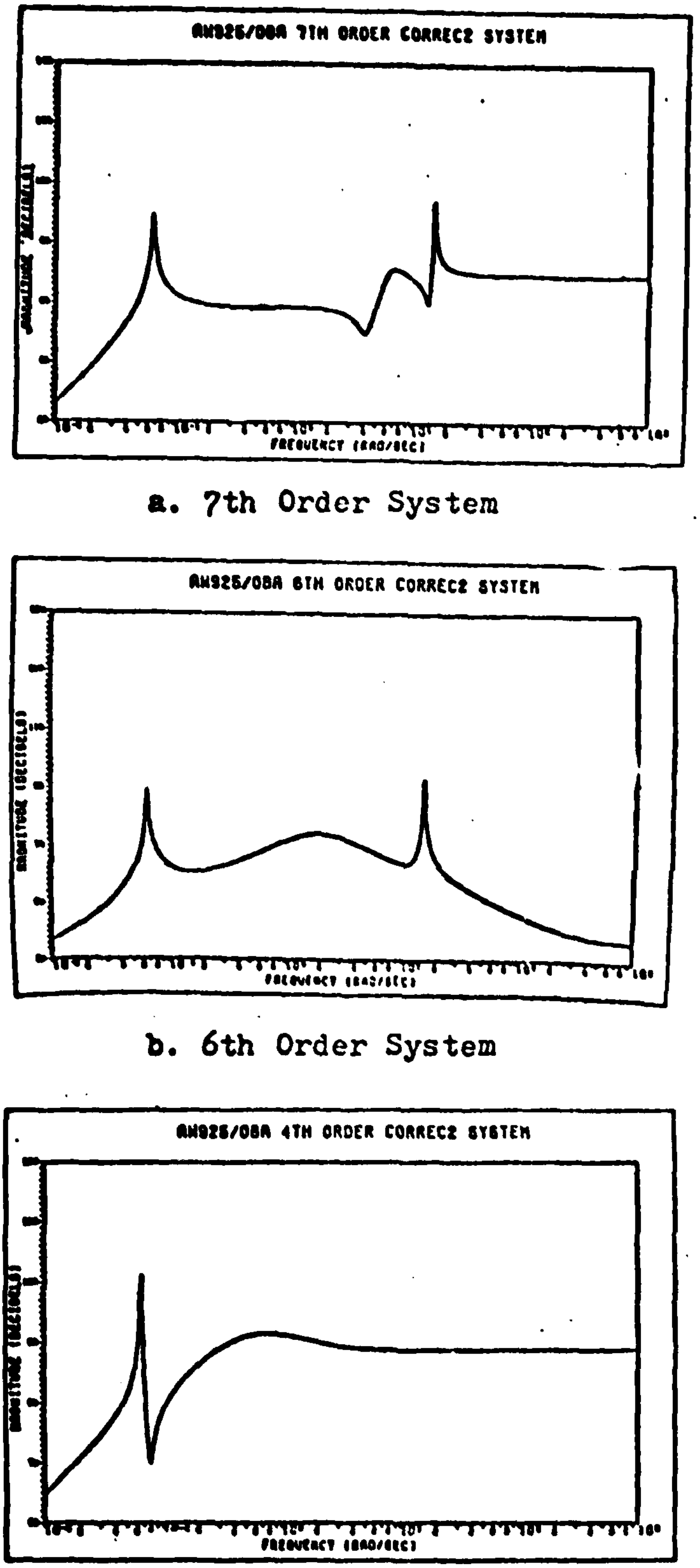


a. 7th Order System

b. 6th Order System

c. 4th Order System

FIGURE 29. AW925/OBA To .1 Rad. Step Input Using Step Balancing

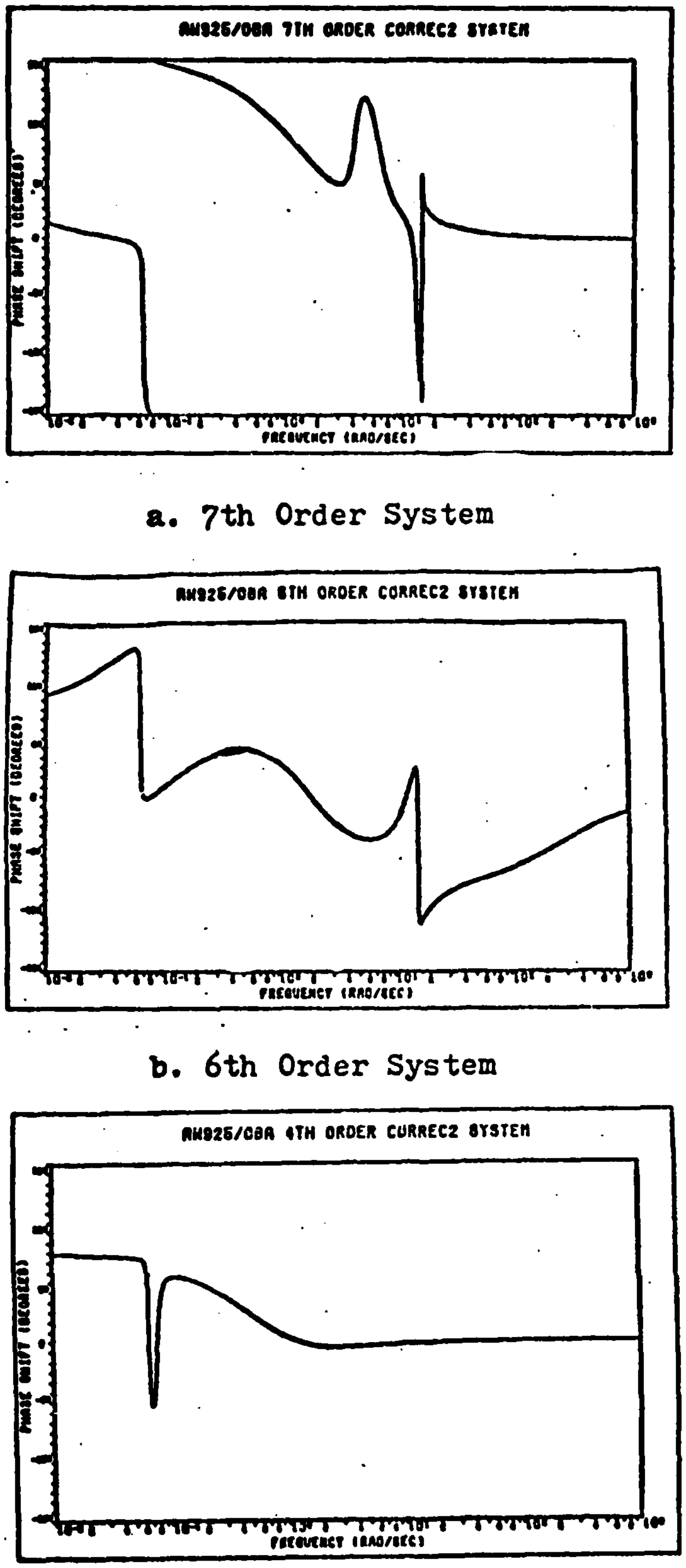


a. 7th Order System

b. 6th Order System

c. 4th Order System

FIGURE 30. Frequency Response (AW925/OBA) Using Step Balancing



a. 7th Order System

b. 6th Order System

c. 4th Order System

FIGURE 31. Phase Response (AW925/OBA) Using Step Balancing

The algorithm is easily programmable and the highly efficient subroutine packages for singular value decomposition are readily available and numerically very stable (Ref 13, 36). Potential applications of the Moore algorithm include simulation, on-line control, and Kalman filter design. The derivation of a reduced order model requires no human trial and error process.

The H_{INF} matrix gives a general guideline in the choice of the minimum order for the reduced order model to "accurately" reproduce the original, full-order system.

Although the steady state controllability and observability grammians, which were used exclusively for this study, only allow application of the Moore algorithm to stable, time-invariant systems, implementation of the finite interval grammians would allow application of the Moore algorithm to unstable and time-varying systems.

The Moore algorithm can be used when other classes of inputs are utilized. The lowest achievable order for inputs other than an impulse or step will probably not be as low as if the inputs had been an impulse or step however. This is because the algorithm is "optimized" for impulse and step inputs.

Properties of the Moore algorithm presented in Section II, this section, Appendix B, and References 24 and 25 make the "internally balanced" state coordinate system highly desirable. The internally balanced state coordinate system is the decomposition of Kalman achieved by using "working subspaces" provided by the controllability and observability grammians. The states are ordered with respect to their controllability and observability properties.

Finally, the Moore algorithm may never be the best model order reduction procedure that can be used, but it does provide a fast and accurate determination of a reduced order model requiring no human trial and error process.

VI. RESULTS AND RECOMMENDATIONS

The goals of this study were twofold. First, the desire was to investigate the Moore algorithm through its application to the relatively high order B-52E CCV flutter control problem. Second, it was desired to develop an interactive computer program which enables a user to not only obtain the balanced representation (via the Moore algorithm) for his system, but provides other options useful in the study of multi-input-multi-output systems as well. This section describes to what extent these objectives were met and the results thereof, as well as recommends areas for future investigation.

SUMMARY OF RESULTS

There are four main results of this study. These results include: successful determination of a tenth order model via the Moore algorithm to compare with AFFDL's tenth order model for the B-52E CCV flutter control problem; successful investigation of the singular values of the H_{INF} matrix as a means of determining the minimum dimension of a reduced order system for "accurate" full order system reproduction; and a working, interactive computer program together with a fully documented user's manual for the program. Each result will now be discussed individually.

Comparison of reduced order models for B-52E CCV flutter control problem. In Section V the Moore algorithm was applied to the B-52E CCV flutter control problem's 24th order model. Figures 23, 24 showed the tenth and ninth order systems obtained via the Moore algorithm vs. the tenth order system obtained via the modified Schwendler and MacNeal technique used by AFFDL. The results are good. The phase shift and damping changes observed in AFFDL's response are not present in the Moore algorithm responses. Since the definition of "goodness" is usually user-defined, it is up to the reader to determine which technique provides the better results. At the very least, however, the Moore algorithm offers desirable properties that the Schwendler and MacNeal technique does not. It is fast, accurate and efficient where AFFDL's technique may be more accurate (dependent on user definition), but is definitely much slower in terms of user development time than the Moore algorithm.

Each technique has its own advantages and disadvantages. AFFDL's technique deals with the "physical" state coordinate system and thus results may be interpreted in a physical way. The Moore algorithm works solely in a mathematical coordinate system and thus no such physical interpretations can be made.

The Moore technique also offers a guideline to the choice of minimum order for a reduced order model. The Moore technique is applicable to linear, time-invariant, stable, MIMO, or SISO systems. The technique could be applied to unstable systems as well, by either:

1) numerically integrate to solve for the controllability and observability grammians over finite time intervals.

2) obtain an algorithm which solves the two matrix Ricatti equations for an A matrix with negative and positive eigenvalues. These matrix Ricatti equations are (Ref 20, 24, 25):

$$A\,W^{a}_{c,\infty} + W^{a}_{c,\infty}\,A^{t} = -BB^{t} \qquad (73)$$

$$A^{t}\,W^{a}_{o,\infty} + W^{a}_{o,\infty}\,A = -C^{t}C \qquad (74)$$

The Moore algorithm may possibly never be the best of all possible algorithms that exist, but it does provide an accurate reduced order model swiftly and efficiently.

H_{INF} matrix singular values investigation. Section V presented the singular values of the H_{INF} matrix for both impulse balancing and step balancing. The results showed that the H_{INF} matrix singular values do give a fairly accurate guideline to follow in determining the minimum acceptable order of the reduced order system.

MIMO -- an interactive computer program. Section III presented the interactive computer program MIMO. Although time did not allow the development of an extensive error checking capability, the program does function as designed and offers fifteen options that aid a knowledgeable user in the analysis of multi-input-multi-output systems.

The computer program offers the following options:

0) List all available options.

1) Terminate MIMO--update local file MEMORE.

2) Input system. Obtain impulse balanced or step balanced coordinate system.

3) Discretize the input system. Obtain F,G, controllability, observability, and Hankel matrices.

4) Plot and optionally list singular values and their ratios vs. sample time for controllability matrix, observability matrix, and/or Hankel matrix.

5) Obtain continuous time controllability and observability matrices.

6) Estimate the condition number of a square matrix.

7) Obtain singular values and/or right and left singular vectors of a matrix.

8) Plot and optionally list the frequency response of user-specified order of balanced model.

9) Dispose plots to AFIT plotter.

10) Update local file MEMORE.

11) Recover information from MEMORE.

12) Interface to TOTAL.

13) Obtain output normal state coordinate system.

14) Obtain output predictive state coordinate system.

15) Obtain the controllability and observability grammians for the input system.

User's manual. The user's manual is an important part of the development of the computer program. Examples were included to illustrate the program's options. The explanations of each option were made as clear as possible bearing in mind the information to be presented. The assumption was made that the potential user knew how to login to the interactive terminal. Beyond that assumption, the user's manual explains all other steps up until logout. The user's manual is found in Appendix C.

RECOMMENDATIONS FOR FUTURE STUDY

There are several ways that this study can be expanded upon. Each recommendation is now presented.

As mentioned in Section V, the Moore algorithm obtains a system which is robust to perturbations in system elements. Therefore, a useful and interesting follow-on would be to create model mismatch by perturbing the eigenvalues of the system by given percentages. The results should indicate a correlation between the condition number and the amount of model mismatch that can be tolerated for a particular system design.

Another interesting application would be to extend the algorithm to discrete-time systems. The continuous time H_{INF} matrix could possibly be replaced by the finite dimension discrete-time Hankel matrix in the existing algorithm. The two grammians that must be solved would then have to be numerically integrated. Other changes would also probably have to be made.

Many other options could be added to MIMO. With the complexity of systems today, an interactive computer package is a must. New routines could be highly useful in the analysis and design of a given system. Also, an efficient error checking capability coupled with an improved user interface would be highly beneficial.

The Moore algorithm as currently programmed is not as efficient in terms of computer resources required, as possible. The efficient coding of this algorithm by someone well versed in that area for potential on-line applications would be a very desirable follow-on. Analyses could then be done to determine which algorithm out of a field of algorithms had the most potential for on-line applications. These are but a few of the areas that could be investigated.

Since there is a great need for efficient model reduction algorithms, it is hoped that the results of this study will be useful in the determination of such algorithms.

BIBLIOGRAPHY

1. Anderson, L.R. Decoupling and Order Reduction of Two Time-Scale Control Systems. Aerospace and Ocean Engineering Dept. Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

2. Antoh, H. Elementary Linear Algebra. New York: John Wiley & Sons Inc., 1977.

3. Appiah, R.K. "Linear Model Reduction Using Hurwitz Polynomial Approximation", International Journal of Control, Vol. 28, No. 3, 1978.

4. ASD Computer Center. ASD Computer Center Calcomp Plotter Guide (Revision A). Wright-Patterson AFB, Ohio: ASD Computer Center 1976.

5. ASD Computer Center. ASD Computer Center CDC NOS/BE User's Guide (RevisionA). Wright-Patterson AFB, Ohio: ASD Computer Center, 1976.

6. Chidambarra, M.R. "On a Method for Simplifying Large Linear Dynamic Systems", IEEE Tran. on Auto. Control. Vol. AC-12, February 1967.

7. Cline, A.K. et al. "An Estimate for the Condition Number of a Matrix", Siam Journal of Numerical Analysis. Vol. 16, No. 2, 1979.

8. Control Data Corporation. Fortran Extended Version 4 Reference Manual (Revision B). Sunnyvale, California: Control Data Corporation, 1976.

9. Control Data Corporation. Intercom Version 4 Reference Manual (Revision B). Sunnyvale, California: Control Data Corporation, 1976.

10. Control Data Corporation. Loader Version 1 Reference Manual (Revision J). Sunnyvale, California: Control Data Corporation, 1976.

11. Dazzo, J.J., and C.H. Houpis. Linear Control System Analysis and Design: Conventional and Modern. New York: McGraw-Hill Book Co., 1975.

12. Derusso, P.M., R.J. Roy, and C.M. Close. State Variables for Engineers. New York: John Wiley & Sons Inc., 1965.

13. Dongarra, J.J. et al. LINPACK User's Guide. Philadelphia: Siam, 1979.

14. El-Attar, R.A., and M. Vidyasagar. "Order Reduction by l_1 and l_∞-Norm Minimization", IEEE Trans. Auto. Control, Vol AC-23, No. 4, 1978.

15. Forsythe, G., Malcom, M.A., and C.B. Moler. Computer Methods for Mathematical Computations. Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

16. Gantmacher, F.R. The Theory of Matrices. Chelsea Publishing Co., New York, 1964.

17. Hutton, M.F., and B. Friedland. "Routh Approximations for Reducing Order of Linear, Time-Invariant Systems", IEEE Trans. Auto. Control, Vol. AC-20, No. 3, 1975.

18. Hutton, M.F., and M.J. Rabins. "Simplification of High-Order Mechanical Systems Using the Routh Approximation", Journal of Dynamic Systems, Measurements, and Control, Vol. 97, 1975.

19. Kalman, R.E. "Mathematical Description of Linear Dynamical Systems", J. SIAM Control, Vol. 1, No. 2, 1963.

20. Kleinman, D.L. A Description of Computer Programs for Use in Linear Systems Studies. TR-77-2. Storrs, Connecticut: School of Engineering, University of Connecticut, 1977.

21. Larimer, S.J. An Interactive Computer Aided Design Program for Digital and Continuous Control System Analysis and Synthesis. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1978. (AD A055 418)

22. Maybeck, P.S. Professor, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (Personal interview), October 31, 1979.

23. Melsa, J.L., and S.K. Jones. Computer Programs for Computational Assistance in the Study of Linear Control Theory (Second Edition). New York: McGraw Hill Book Co., 1973.

24. Moore, B.C. Singular Value Analysis of Linear Systems, Part I: External Variables. Systems Control Report No. 7801, Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario. (1978)

25. Moore, B.C. Singular Value Analysis of Linear Systems, Part II: Controllability, Observability, and Model Reduction. Systems Control Report No. 7802, Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario. (1978)

26. O'Brien, F.L. A Consolidated Computer Program for Control System Design. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1976. (AD A034 879)

27. Reddy, A.S.S.R. "A Method for Frequency Domain Simplification of Transfer Functions", International Journal of Control, Vol. 23, 1976.

28. Reid, J.G. Linear Systems Analysis and Digital Computation Methods. (Unpublished Course Notes) Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology.

29. Reid, J.G. Assistant Professor, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (Personal interview) April 17, 1979.

30. Reid, J.G, et al. "Robustness Properties of Output Predictive Dead-Beat Control: SISO Case". To be published. Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (1979)

31. Schwanz, R.C. Analytical Flight Test Measurements for Estimation of Aerodynamic Parameters of Flexible Aircraft. AFFDL TM-78-104-FGC. Wright-Patterson AFB, Ohio: AF Flight Dynamics Laboratory, 1978.

32. Schwendler, R.G., and R.H. MacNeal. Optimum Structural Representation in Aeroelastic Analyses. ASD-TR-61-680. Wright-Patterson AFB, Ohio: Flight Control Laboratory, 1962.

33. Shamash, T. "Model Order Reduction Using the Routh Stability Criterion and the Pade' Approximation Technique", International Journal of Control. Vol. 21, No. 3, 1975.

34. Shieh, L. S., and M. J. Goldman. "A Mixed Cauer Form for Linear System Reduction", IEEE Tran. On Systems, Man, and Cybernetics. November 1974.

35. Skelton, R. E. "Observability Measures and Performance Sensitivity in the Model Reduction Problem", International Journal of Control. Vol. 29, No. 4, 1979.

36. Stewart, G. W. Introduction to Matrix Computations. New York: Academic Press, 1973.

APPENDIX A

DEFINITION OF THE SINGULAR VALUE DECOMPOSITION

APPENDIX A

DEFINITION OF THE SINGULAR VALUE DECOMPOSITION

A singular value decomposition occurs when a general matrix is reduced to diagonal form by premultiplying and postmultiplying it by unitary matrices (Ref 36: 318-325, Ref 13: 11.1-11.3, Ref 24: 6-8). The equation is written as:

$$V^H A_{mxp} U = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{; where } V^H \text{ denotes the conjugate transpose of } V \qquad \text{(A-1)}$$

where A is a general matrix

$\Sigma_1 = \text{diag } (\sigma_1, \sigma_2, \ldots \sigma_N)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$

$n > r$

$r = \min (m, p)$

V, U are unitary matrices

The columns of V are called the left singular vectors. They are the eigenvectors of AA^H. The columns of U are called the right singular vectors. They are the eigenvectors of A^HA. The singular values; $\sigma_1, \sigma_2, \ldots \sigma_r$, are the non-negative square roots of the eigenvalues of A^HA.

Thus, any general matrix can be factored into:

$$A_{mxp} = V_{mxm} \Sigma_{mxp} U^H_{pxp} \qquad \text{(A-2)}$$

This factorization of a matrix has several "nice" properties. Since only unitary matrices are used, errors are not magnified. Thus, the rxr sub-matrix Σ_1 will only be of defective rank if the matrix is of defective rank (Ref 36:318).

The left singular vectors are the axes of an ellipsoid with the corresponding singular values being the lengths of the axis of the ellipsoid. Thus $\sigma_1 = \sigma_{max}$ is the length of the major axis, and $\sigma_r = \sigma_{min}$ is the length of the minor axis of the ellipsoid.

Defining a quantity, K(A), as the ratio of σ_{max} to σ_{min} ($\sigma_{max}/\sigma_{min}$), we have a measure of eccentricity of the ellipsoid. The scalar K(A) is termed the "condition number with respect to inversion" of the matrix A. The larger in magnitude the condition number, the more eccentric (near degenerate) is the ellipsoid, and thus, the more closely the A matrix is to being singular. Therefore, unlike eigenvalues which tell us only if a matrix is singular or not, singular values give us an idea of how close we are to being singular (i.e., how big the perturbations in the elements of the matrix can be before the matrix becomes singular).

Defining $\|A\|_2 = \max\limits_{\|\underline{x}\|_2 = \|\underline{y}\| = 1} |\underline{y}^T A\underline{x}|$

(A-3)

as the 2- norm of a matrix (Ref 36:180), $\sigma_1 = \sigma_{max}$ is indeed the value of the norm; i.e., $\| A\| = \sigma_1$. Likewise, $\|A^+\| = {}^1/\sigma_r$; where A^+ denotes the generalized inverse. (A-4)

Finally, $K(A) = \sigma_{max}/\sigma_{min}$ also equals $\|A\| \quad \|A^+\|$, which is the formal definition of K(A) (Ref 7:1).

Singular values have many other properties which are important, but are beyond the scope of this appendix. The reader is referred to Reference 36:318-325. The material presented in this appendix hopefully will acquaint the reader who has a linear algebra background with enough of the basics of the singular value decomposition to follow the development in the text.

APPENDIX B

DESCRIPTION OF THE ALGORITHM

(Ref 24, 25)

APPENDIX B

DESCRIPTION OF THE ALGORITHM
(Ref 24,25)

The algorithm to be presented here was developed by Dr Bruce Moore of the University of Toronto. The algorithm will be developed and presented to enhance the understanding of the properties and results from the application of this algorithm. A more extensive development may be found in References 24 and 25.

To comprehend the development of this algorithm, one requires an understanding of singular value analysis. A brief description of singular value analysis is presented in Appendix A. The reader is also referred to Reference 36 for an excellent treatment of the subject.

Singular value analysis is applied to linear static equations, followed by its application to linear differential equations. The results will then be applied to state space models yielding the "internally balanced" representation, which will result in reduced order models. Results for the algorithms as applied to a third order example as described by El-Attar and Vidyasagar in Reference 14 are presented in Section IV of this thesis.

The following discussion is divided into two major areas (Ref 24,25):

1) external variable analysis (those which may be measured and/or manipulated),

2) internal variable analysis (state variable coordinate system). Both sets of variables need to be considered for an effective model reduction technique.

The notation will be the same as Dr. Moore used in References 24 and 25 to enable the reader to easily make the transition between this paper and Dr. Moore's papers. The notation is:

R	The field of real numbers
$C[0, t_1]$	ring of piecewise continuous functions on interval $[o, t_1]$
R^m	m^{th} dimensional vector space
$C^m[0, t_1]$	m^{th} dimensional vector space
Ker(M)	Null space of M
Im(M)	Range space of M
M^T	transpose of M
S	Subspace
$S^{\perp}$, δ R^N	orthogonal complement in R^N
Capital letters	Matrices and maps
Underscored letters	Vectors
$\|\underline{v}\|$	Euclidean norm of a vector
$\|M\|$	Subordinate Spectral norm for a matrix

where $$\|M\| = \max_{\|\underline{x}\| = 1} \|M\underline{x}\|$$

For the linear equation

$$\underline{n}_r = M_{r \times m} \underline{w}_m \tag{B-1}$$

where M can be thought of as a map transforming R^m into R^r (Ref 28:77-78), there are four subspaces which completely characterize equation (B-1). These subspaces are:

$$\mathrm{Im}(M)$$
$$\mathrm{Im}(M)^{\perp} = \mathrm{Ker}(M^T)$$
$$\mathrm{Ker}(M)$$
$$\mathrm{Ker}(M)^{\perp} = \mathrm{Im}(M^T).$$

For a non-trivial solution ($\underline{W} \neq \underline{0}$) to exist, Ker(M) $\neq 0$. The columns of M must be linearly dependent (Ref 28:3-71). Then by defining V_1 and V_2 to be matrices with orthogonal columns (basis vectors) which span Ker(M) and Ker(M), respectively, we may write the vector $\underline{w}$ as

$$\underline{w} = V_1 \underline{\hat{w}}_1 + V_2 \underline{\hat{w}}_2. \tag{B-2}$$

Substituting equation (B-2) into equation (B-1) and rewriting yields

$$\underline{n} = MV_1 \underline{\hat{w}}_1 + MV_2 \underline{\hat{w}}_2 \tag{B-3}$$

Since Ker(M) is comprised of the subspace of all possible solutions to the homogeneous linear equation

$$M\underline{w} = \underline{0} \text{ (Ref 28:3-58),} \tag{B-4}$$

$MV_2 = \underline{0}$. Therefore, equation (B-2) reduces to

$$\underline{n} = MV_1 \underline{w}_1 \tag{B-5}$$

Thus, it is seen that the variable $\hat{\underline{w}}_2 = V_2^T \underline{w}$ has no effect on the system. A redundancy exists and is stripped away by reduction of equation (B-1) to equation (B-5).

Similarly, if we define U_1, U_2 as two matrices whose orthonormal columns span Im(M) and $\text{Im}(M)^{\perp}$, respectively, then equation (B-1) can be rewritten as

$$U_1^T, \underline{n} + U_2^T \underline{n} = U_1^T M\underline{w} + U_2^T M\underline{w}, \tag{B-6}$$

which reduces to

$$U_1^T \underline{n} = U_1^T M\underline{w}; \text{ where } U_2^T \underline{n} = \underline{0} \tag{B-7}$$

This exposes the redundancy associated with $\underline{n}$.

Combining the results of equations (B-5) and (B-7), thereby removing the redundancy yields

$$U_1^T \underline{n} = (U_1^T MV_1)\, \hat{\underline{w}}_1 . \tag{B-8}$$

The minimum norm vector $\underline{w}$ which minimizes $\|\underline{R} - M\underline{w}\|$ is

$$\underline{w} = V_1 \hat{\underline{w}} , \tag{B-9}$$

where $\hat{\underline{w}}$ satisfies

$$U_1^T \underline{n} = (U_1^T M V_1) \hat{\underline{w}} \tag{B-10}$$

Unfortunately, though theoretically sound, the above computations might yield erroneous results when programmed into a digital computer.

Fixed wordlengths, truncation and the like can cause a matrix of less than full rank to appear to be of full rank. However, through use of the singular value decomposition these restrictions' effects are greatly reduced.

If we define the following subspaces

$$\begin{aligned} S_m &= \{\underline{n} \in R^r : \underline{n} = M\underline{w}, \ \|\underline{w}\| = 1\} \\ S_m^T &= \{\underline{w} \in R^m : \underline{w} = M^T\underline{n}, \ \|\underline{n}\| = 1\} \end{aligned} \tag{B-11}$$

which are contained in IM(M), Ker(M), respectively, and perturb M by no more than $\|\Delta M\|$ ($M_\Delta = M+\Delta M$), then $S_{m\Delta}$, $S^T_{m\Delta}$ must be "nearly" in those subspaces.

Referring to Appendix A, the singular value factorization of a matrix M yields

$$M = U_1 \Sigma_1 V_1^T . \tag{B-12}$$

U_1 and V_1 have orthonormal columns which span Im(M) and $\text{Ker}(M)^\perp$ respectively.

Then, by defining U_2, V_2 to be matrices with orthonormal columns which span $\mathrm{Im}(M)^{\perp}$, $\mathrm{Ker}(M)$, respectively, the singular value factorization of M may be rewritten as

$$M = (U_1 U_2) \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U\Sigma V^T. \tag{B-13}$$

The columns of U are the left singular vectors, while the columns of V are the right singular vectors. We can rewrite $\underline{n} = M\underline{w}$ now as

$$\begin{bmatrix} \underline{\bar{n}}_1 \\ \underline{\bar{n}}_2 \end{bmatrix} = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \quad M\underline{w} = \begin{bmatrix} \Sigma_1 \; V_1^T \; \underline{w} \\ 0 \end{bmatrix} \tag{B-14}$$

where $\underline{\bar{n}}_1 = U_1^T \underline{n}_1$

$\underline{\bar{n}}_2 = U_2^T \underline{n}_2$.

Thus we can obtain

$$\Sigma_1^{-1} \underline{\bar{n}}_1 = V_1^T \underline{w}$$

$$\| \Sigma_1^{-1} \underline{\bar{n}}_1 \|^2 = \| V_1^T \underline{w} \|^2 \leq \| \underline{w} \|. \tag{B-15}$$

Equation (15) allows us to view the singular value factorization from a geometrical point-of-view. For, every vector in S_m is contained an elliptical region in R^r. The boundary of which is defined by (Ref 24:7)

$$\frac{\bar{n}^2_{1,1}}{\sigma^2_{m,1}} + \ldots + \frac{\bar{n}^2_{1,\ell}}{\sigma^2_{m,\ell}} = 1. \qquad \text{(B-16)}$$

where

$$\sigma_{m,i} \geq \sigma_{m,i+1}$$

As we saw in equations (B-6) and (B-7), U_2 is associated with redundancy as applied to $\underline{n} = M\underline{w}$. Therefore, the column vectors of U_2 are unit vectors aligned with a degenerate axis of an ellipse, where its corresponding singular value is the length of that axis. Just the opposite, the columns of U_1 are the nondegenerate axes of the ellipses.

Hence, the degenerate axis of each ellipse even under slight perturbation spans a space very close to $\text{Im}(M)^{\perp}$. The nondegenerate axis of each ellipse spans a space very close to the Im(M).

Similarly, the subspace S^T_m can be so characterized. Let the columns of V be the coordinate vectors ϵR^m, so that $\underline{n} = M\underline{w}$ becomes

$$\underline{n} = M(V_1 V_2) \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \end{bmatrix} = U_1 \Sigma_1 \hat{w}_1 . \qquad \text{(B-17)}$$

We can write

$$\| \Sigma_1 \hat{w}_1 \|^2 = \| U_1^T \underline{n} \|^2 = 1 \qquad \text{(B-18)}$$

Thus the equation of the ellipse can now be written as (Ref 24:8)

$$\frac{\hat{w}_{1,1}^{2}}{(1/\sigma_{m,1})^{2}} + \ldots\ldots + \frac{\hat{w}_{1,\ell}}{(1/\sigma_{m,\ell})^{2}} = 1 \qquad \text{(B-19)}$$

From Appendix A we know that

$$\sigma_1 = q_{max} = \|M\|. \qquad \text{(B-20)}$$

Using the equation $M_\Delta = M+\Delta M$, we can define an upperbound on the effect of a perturbation on M by writing

$$|\sigma_{m,1} - \sigma_{m_\Delta,1}| = \|\Delta M\| = \sigma_{\Delta m,\,max}\,. \qquad \text{(B-21)}$$

Then generalizing, equation (B-21) can be rewritten as

$$|\sigma_{m,i} - \sigma_{m_\Delta,i}| \le \|\Delta M\|. \qquad \text{(B-22)}$$

Singular values are also useful in the determination of the worst case sensitivity of minimum norm solutions of $\underline{n} = M\underline{w}$ to perturbations in $\underline{n}$.

Defining $\underline{w}, \underline{w}_\Delta$ to be minimum norm vectors satisfying $\underline{n} = M\underline{w}$, such that the matrices U and V in the singular value factorization of M are the Identity matrix, the following equation can be written.

$$\frac{\|\Delta\underline{w}\|}{\|\underline{w}\|} = \frac{\sigma_{m,1}}{\sigma_{m,\ell}} \; \frac{\|\Delta\underline{n}\|}{\|\underline{n}\|}\,, \qquad \text{(B-23)}$$

where if M is nxn, $\sigma_{m,1}/\sigma_{m,\ell}$ is the condition number with respect to inversion (Ref 24:9).

The previous results show us that though the singular value factorization of M is

$$M = U_1 \Sigma_1 V_1^T = [U_{1,L}\ U_{1,s}] \begin{bmatrix} \Sigma_{1,L} & 0 \\ 0 & \Sigma_{1,s} \end{bmatrix} \begin{bmatrix} V_{1,L}^T \\ V_{1,s}^T \end{bmatrix}, \tag{B-24}$$

where L denotes "large", and s denotes "small", can be rewritten as

$$M = U_1 \Sigma_1 V_1^T = U_{1,L} \Sigma_{1,L} V_{1,L}^T \tag{B-25}$$

if the elements in $U_{1,s}\ \Sigma_{1,s}\ V_{1,s}^T$ are smaller in magnitude than the uncertainty of the given system.

Now, expanding the previous results to systems of linear differential equations, many of the properties apply. The equation of concern is

$$\underline{n}(t) = M(t)\, \underline{w}(t),\ 0 \leq t \leq t_1 . \tag{B-26}$$

The map $H_{m,t_1} : R^m \rightarrow C^r(0, t_1)$ is defined by equation (B-26). The map $L_{m,t_1} : C^m(0,t_1) \rightarrow R^r$ is defined by

$$\underline{n}(t_1) = \int_0^{t_1} M(t-\tau)\, \underline{U}(\tau)\, d\tau. \tag{B-27}$$

Again, associated with the map M(t) the following fundamental subspaces exist:

$$\mathrm{Im}\,(L_{m,t_1}) = \{\,\underline{n} : \underline{n} = \int_0^{t_1} M\,(t_1 - \tau)\,\underline{U}(\tau)d\tau, \underline{U} \epsilon C^m(0, t_1)\,\} \tag{B-28}$$

$$\mathrm{Ker}\ (H_{m,t_1}) = \{\,\underline{w} : M(t)\,\underline{w} = 0,\ 0 \le t \le t_1\,\}$$

$$\mathrm{Im}\,(L_{m,t_1})^{\perp} = \mathrm{Ker}\ (H_m{}^T, t_1)$$

$$\mathrm{Ker}\ (H_{m,t_1})^{\perp} = \mathrm{Im}\,(L_m{}^T, t_1)$$

As before, if $\mathrm{Im}(L_{m,t_1})$ does not span R^r (such that a non-trivial solution does exist), then there exists two matrices U_1, U_2 with orthonormal columns which span $\mathrm{Im}(L_{m,t_1})$, $\mathrm{Im}(L_{m,t_1})^{\perp}$ so that

$$\int_0^{t_1} U_2^T M(t_1 - \tau)\,\underline{U}\,(\tau)\,d\tau = \underline{0} \text{ for all } \underline{U} \epsilon C^m(0, t_1).$$

Rewriting equations (B-26) and (B-27) yields

$$U_1^T \underline{n}(t) = U_1^T M(t)\,\underline{w} \qquad 0 \le t \le t_1 \tag{B-29}$$

$$U_2^T \underline{n}(t) = \underline{0}$$

and

$$U_1^T \underline{n}(t_1) = \int_0^{t_1} U_1^T M\,(t_1 - \tau)\,\underline{U}\,(\tau)d\tau \tag{B-30}$$

$$U_2^T \underline{n}\,(t_1) = \underline{0}.$$

Similarly, if Ker $(m, t_1) = 0$, then two matrices V_1, V_2 exist which span Ker $(Hm, t_1)^{\perp}$ and Ker(Hm, t_1), respectively. Writing

$$\underline{w} = V_1 \underline{\hat{w}}_1 + V_2 \underline{\hat{w}}_2 \qquad \text{(B-31)}$$
$$\underline{U}(t) = V_1 \underline{\hat{U}}_1(t) + V_2 \underline{\hat{U}}_2(t),$$

where $M(t)V_2 = 0$ for $t \epsilon [0, t_1]$, equations (B-26) and (B-27) can be rewritten as

$$\underline{n}(t) = M(t) V_1 \underline{\hat{w}}_1, \; 0 \le t \le t_1 \qquad \text{(B-32)}$$
$$\underline{n}(t_1) = \int_0^{t_1} M(t_1 - \tau) V_1 \underline{\hat{U}}_1(\tau) d\tau.$$

Combining equations (B-29) and (B-32) yields

$$U_1^T \underline{n}(t) = (U_1^T M(t) V_1) \underline{\hat{w}}_1, \; 0 \le t \le t_1 \qquad \text{(B-33)}$$
$$U_1^T \underline{n}(t_1) = \int_0^{t_1} (U_1^T M(t_1 - \tau) V_1) \underline{\hat{U}}_1(\tau) d\tau$$

Therefore, in order to use the results of the previous section, and to be able to program the algorithm on a computer, we need matrix representations of the four subspaces.

Forming the equation

$$W^2_{m, t_1} = \int_0^{t_1} M(t) M^T(t) dt \text{ (Grammian)}, \qquad \text{(B-34)}$$

that important matrix representation is achieved. This matrix is W_{m, t_1}, which is the symmetric square root of equation (B-34).

Then $\text{Im}(Lm, t_1) = \text{Im}(Wm, t_1)$, and $\text{Ker}(Hm, t_1) = \text{Ker}(Wm^T, t_1)$. Application of singular value analysis to $Wm, t_1 (Wm^T, t_1)$ will reflect the norm characteristics of the maps with no distortion.

This allows the geometric concept obtained previously to be extended to linear differential equations. The reader should note that Wm, t_1 is symmetric and positive definite. Similar to equation (B-22), equation (B-35) can be written as

$$|\sigma_{1,i}^2 - \sigma_{2,i}^2| \le \left\| \int_0^{t_1} \Delta M^T \Delta M(t)\, dt \right\| . \qquad \text{(B-35)}$$

To define the worst case magnification of perturbation in the solution of $\underline{n}(t) = M(t)\underline{w}(t)$, the condition numbers of Wm, t_1 and Wm^T, t_1 may be used.

Moore shows in Reference 24 that if $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_r \ge 0$ (singular values of Wm, t_1), $n \in R^r$, then $\underline{U}n \in C^m(0, t_1)$ be the minimum norm function satisfying $\underline{n} = Lm, t_1(\underline{U}_n)$, produces

$$\left[\frac{\int_0^{t_1} \| \Delta \underline{U}(t) \|^2 dt}{\int_0^{t_1} \| \underline{U}_n(t) \|^2 dt}\right]^{\frac{1}{2}} = \left[\frac{\sigma_1}{\sigma_r}\right] \frac{\| \Delta \underline{n} \|}{\| \underline{n} \|} , \qquad \text{(B-36)}$$

where

$$\Delta \underline{U}(t) \equiv \underline{U}(t)_{n+\Delta n} - \underline{U}_n(t) .$$

Similarly, Moore shows that given $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0$ (singular values of Wm^T, t_1), there exist $\underline{w}$, $\Delta\underline{w}$ such that

$$\left[\frac{\int_0^{t_1} \| M(t)\ \Delta\ \underline{w}\ \|^2 dt}{\int_0^{t_1} \| M(t)\ \ \underline{w}\ \|^2\, dt}\right]^{\frac{1}{2}} = \left[\frac{\sigma_m}{\sigma_1}\right] \ \frac{\| \Delta\, \underline{w}\ \|}{\| \ \underline{w}\ \|} \quad . \tag{B-37}$$

Extending the previous results of linear static equations once again allows the equations

$$\begin{aligned} W_{m\Delta,\,t_1} &= (\hat{U}_1\, \hat{U}_2) \begin{bmatrix} \hat{\Sigma}_L & 0 \\ 0 & \hat{\Sigma}_S \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \\ W_{m^T\Delta,\,t_1} &= (\tilde{V}_1\, \tilde{V}_2) \begin{bmatrix} \tilde{\Sigma}_L & 0 \\ 0 & \tilde{\Sigma}_S \end{bmatrix} \begin{bmatrix} \tilde{V}_1^T \\ \tilde{V}_2^T \end{bmatrix} \end{aligned} \tag{B-38}$$

to be written. If $\tilde{\Sigma}_S$, $\tilde{\Sigma}_S$ elements are small, $\hat{U}_1$, $\hat{U}_2$, $\tilde{V}_1$, $\tilde{V}_2$ may be used as working subspaces for $\mathrm{Im}(Lm, t_1)$, $\mathrm{Im}(Lm, t_1)^{\perp}$, $\mathrm{Ker}(Hm, t_1)^{\perp}$, and $\mathrm{Ker}(Hm, t_1)$, respectively.

The previous results will now be applied to external variables (those which may be manipulated and/or measured). Here, we can define the system to be $\underline{Y}(t) = M(t)\underline{U}(t)$.

Defining $Wm, t_1 = U_{out}\, \Sigma_{out}\, U_{out}^T$ and $Wm^T, t_1 = V_{in}\, \Sigma_{in}\, V_{in}^T$ (where "out" and "in" suggest association with outputs and inputs), we may apply orthogonal transformations to yield

$$\hat{\underline{Y}}(t) = U_{out}^{T}\,\underline{Y}(t), \quad \underline{U}(t) = V_{in}\,\hat{\underline{U}}(t). \qquad \text{(B-39)}$$

Our system is now $\hat{\underline{Y}}(t) = M(t)\ \hat{\underline{U}}(t)$. Then, $\hat{\underline{Y}}(t) = \hat{M}(t)\,\hat{\underline{U}}(t) = U_{out}^{T}$ $M(t)\,\hat{\underline{U}}(t) = U_{out}^{T}\,M(t)\,V_{in}\,\hat{\underline{U}}(t)$. (B-40)

Therefore, $\hat{M}(t) = U_{out}^{T}\,M(t)\,V_{in}$. Proceeding to obtain the grammians yields

$$\begin{aligned} W_{\hat{m},t_1} &= \int_0^{t_1} \hat{M}(t)\,M^T(t)\,dt = \Sigma^2_{out} \\ W_{\hat{m}^T,t_1} &= \int_0^{t_1} \hat{M}^T(t)\,\hat{M}(t)\,dt = \Sigma^2_{in} \end{aligned} \qquad \text{(B-41)}$$

With this result, we may say that the outputs are ordered with respect to responsiveness to an input, and the inputs are ordered with respect to their influence on the output. Unfortunately, however, Wm, t_1, Wm^T, t_1 are dependent on the internal coordinate system.

Therefore, the discussion will now turn to the internal coordinate system. The discussion will use the state space representation. Given equations of the form

$$\dot{\underline{X}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad \text{(B-42)}$$

where $\underline{Y}(t) = C\,\underline{x}(t)$,

$\underline{X}(t)\ \epsilon R^N$, $\underline{Y}(t)\ \epsilon R^m$.

Let,

$$M_1(t) = e^{At}B, \; M_2(t) = Ce^{At} \qquad \text{(B-43)}$$

Then, $\mathrm{Im}(Lm_1, t_1)$ is the controllable subspace Xc. The unobservable subspace is equal to $\mathrm{Ker}(Hm_2, t_1)$. Then, by previous results we can show that $\mathrm{Im}(Lm_1, t_1) = \mathrm{Im}(Wc, t_1)$, and $\mathrm{Ker}(Hm_2, t_1) = \mathrm{Ker}(Wo, t_1)$. Wc, t_1 is the symmetric square root of $W_c^2, t_1 \equiv \int_0^{t_1} e^{At}BB^Te^{A^Tt}dt$. Wo, t_1 is the symmetric square root of $W_{o,t_1}^2 \equiv \int_0^{t_1} e^{A^Tt}C^TCe^{At}dt$.

Redundancy of state variables is caused by uncontrollable, unobservable modes (Ref 24:25). By applying our geometrical concept provided by singular value analysis to Wc, t_1 and Wo, t_1, we have an "image" of the controllable (observable) subspace. We would like to then interpret near degeneracy of the ellipse (condition number large) as a measure of redundancy of state variables. Unfortunately, as in equation (41) the controllability and observability grammians are state coordinate dependent.

If we apply a coordinate transformation to (A, B, C) to yield (A', B', C'), we have the following equations:

$$\begin{aligned} A' &= T^{-1}AT \qquad \text{(B-44)} \\ B' &= T^{-1}B \\ C' &= CT. \end{aligned}$$

The controllability and observability grammians are related by

$$\hat{W}^{a}_{c,t_1} = T^{-1} W^{a}_{c,t_1} T^{-1^T} \qquad \text{(B-45)}$$

$$\hat{W}^{a}_{o,t_1} = T^{T} W^{a}_{o,t_1} T.$$

Then performing a singular value factorization on Wc,t_1, Wo,t_1 yields

$$W_{c,t_1} = U_{c,t_1} \Sigma_{c,t_1} U^{T}_{c,t_1} \qquad \text{(B-46)}$$

$$W_{o,t_1} = U_{o,t_1} \Sigma_{o,t_1} U^{T}_{o,t_1}$$

The $\cdot,t_1$ notation will now be deleted for simplicity. By letting T^{-1} in equation (B-45) equal $\Sigma_c^{-1} U_c^T$, we get $W_c^a = I$, $W_o^a = H^T H$, where

$$H = \Sigma_o U_o^T U_c \Sigma_c . \qquad \text{(B-47)}$$

The singular values of H are equal to those of W_o. This forces an input normalization.

Let us look at the matrix H in more detail. The continuous time grammians are related to their discrete-time matrix counterparts via

$$W_c^a = \lim_{h \to 0} \frac{1}{h} [M_{con}] [M_{con}]^T \qquad \text{(B-48)}$$

$$W_o^a = \lim_{h \to 0} h [M_{obs}]^T [M_{obs}]$$

where h is sampling time

M_{con} is the discrete-time extended controllability matrix,

M_{obs} is the discrete-time extended observability matrix.

Let W_c, W_o have factorizations $U_c \Sigma_c U_c^T$ and $U_o \Sigma_o U_o^T$, respectively. Let $M_{con} = U_{con} \Sigma_{con} V_{con}T$, and $M_{obs} = U_{obs} \Sigma_{obs} V_{obs}^T$. Moore shows that

$$U_c = \lim_{h \to o} U_{con} \quad \Sigma_c = \lim_{h \to o} \frac{1}{\sqrt{h}} \Sigma_{con} \qquad \text{(B-49)}$$

$$U_o = \lim_{h \to o} U_{obs} \quad \Sigma_o = \lim_{h \to o} \frac{1}{\sqrt{h}} \Sigma_{obs}$$

$$\sigma_H = \lim_{h \to o} \sigma_{HANKEL} \cdot \qquad \text{(B-50)}$$

Equation (B-50) offers an interesting result. The infinite dimensional Hankel matrix ([M_{obs}]) [(M_{con})] <u>is</u> coordinate invariant. This result shows that essential information contained in the Hankel matrix is embodied in the H matrix. The singular values of the Hankel matrix give a measure of closeness to redundancy of state variables. Therefore, the singular value decomposition of Moore's H matrix gives us pertinent information about the redundant state variables. With this H matrix, loosely we have injected controllability/observability properties into a single matrix.

Defining a new system related to the original system via

$$\underline{X} = T\underline{Z} \tag{B-51}$$

where $T = U_O \Sigma_O^{-1} U_H \Sigma_H^{\frac{1}{2}}$ (B-52)

$$H = U_H \Sigma_H V_H^T$$

produces the "internally balanced" state representation. This transformation forces $W_c = W_o = \Sigma_H^{\frac{1}{2}}$. Thus, the internally balanced system is balanced with respect to an "equal" weighting on controllability and observability properties.

Many desirable properties result from the internally balanced representation. Only one of which is the fact that the singular values of the H matrix give us an estimate of how low we can reduce the order of the system without severe loss of accuracy. This can be viewed most easily again from the geometrical point of view. As discussed previously, the H matrix contains information concerning the observability and controllability properties of the particular system. If we observe the singular value of H as

$$\sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, \ldots, \sigma_n, \tag{B-53}$$

where $\sigma_k >> \sigma_{k+1}$, then since the singular values are the lengths of the axes of the hyperellipsoid, can be thought of as being associated with the major axes of the hyperellipsoid. Then, σ_{k+1} through σ_n can be thought of as the minor axes of the hyperellipsoid.

Therefore, if indeed $q_k >> q_{k+1}$, we can view the axes corresponding to σ_{k+1} through σ_n as being degenerate. As mentioned previously, "degeneracy" corresponds to state variable "redundancy" for our coordinate invariant system. Since the H matrix embodies controllability/observability properties, to strip away the states corresponding to σ_{k+1} through σ_n is equivalent to stripping away the state redundancy of the system. Moreover, it is equivalent to stripping away the uncontrollable, unobservable subspace. This is analogous to the Kalman decomposition (Ref 25:20).

Many equations have been presented. However, the development presented by Moore in References 24 and 25 is much more extensive. The reader is encouraged to refer to the material for assistance in grasping the technique.

Section II will take a "step away" from the equations and present the rationale behind the development in a more clear fashion. Also, important properties that this "internally balanced" representation possess, not mentioned here, will be presented. This Appendix was designed to help the interested reader better understand the development of the Moore algorithm, as well as to base the results of Sections III and V upon.

APPENDIX C

USER'S MANUAL FOR MIMO

August 1979

USER'S MANUAL FOR MIMO

This user's manual was prepared by 2nd Lt James R. McClendon, USAF in partial fulfillment of his Masters Thesis at AFIT/ENG, Wright-Patterson AFB, Ohio, August 1979.

```
**************************************************
*                                                *
*                                                *
*  MMMM     MMMM   IIII   MMMM     MMMM  OOOOOOOO *
*  MMMMM   MMMMM    II    MMMMM   MMMMM  OO    OO *
*  MMMMMMMMMMMMM    II    MMMMMMMMMMMMM  OO    OO *
*  MMMMMMMMMMMMM    II    MMMMMMMMMMMMM  OO    OO *
*  MMMMMM MMMMMM    II    MMMMMM MMMMMM  OO    OO *
*  MMMM     MMMM    II    MMMM     MMMM  OO    OO *
*  MMMM     MMMM    II    MMMM     MMMM  OO    OO *
*  MMMM     MMMM    II    MMMM     MMMM  OO    OO *
*  MMMM     MMMM   IIII   MMMM     MMMM  OOOOOOOO *
*                                                *
*                                                *
*                                                *
* ---------------------------------------------- *
*                                                *
*  AN INTERACTIVE COMPUTER PROGRAM CONTAINING    *
*  SEVERAL OPTIONS USEFUL IN CHANGING STATE VARIABLE *
*  COORDINATE SYSTEMS AND PROVIDING REDUCED ORDER *
*  MODELS.                                       *
*                                                *
* ---------------------------------------------- *
*                                                *
*                                                *
*        AIR FORCE INSTITUTE OF TECHNOLOGY       *
*                  AUGUST 1979                   *
*                                                *
*                                                *
**************************************************
```

OVERVIEW OF MIMO

MIMO is designed as a tool for the user interested in model order reduction and singular value analysis of multivariable, linear, time-invariant systems.

MIMO offers a model order reduction algorithm that is highly accurate for both impulse and step response applications.

MIMO provides options to obtain the singular values and the closely related condition number with respect to inversion.

MIMO plots as well as lists the frequency response for a user specified reduced order model.

MIMO also obtains special coordinate systems. These coordinate systems include: the output normal system(states ordered with respect to observability), the output predictive system(where the observability matrix is the identity matrix), and the internally balanced system(which the model order reduction technique is based upon).

MIMO should result in a substantial time savings to the user interested in a reduced order model for his particular system, or interested in investigating the relative importance of states as they effect input/output properties of the system. This qualitative analysis of the states is based heavily upon concepts of singular value analysis(Ref 13,36).

This user's manual underlines everything the user should type in. (Some inputs are circled instead instead for clarity)

MIMO PRELIMINARIES

TO ACCESS MIMO

```
TYPE:  ATTACH MIMO, ID=AFIT
       MIMO
```

The previous two lines will attach the permanent file MIMO. The program will begin execution with the typing of the word MIMO.

PREPARATION

The user should prepare himself with the proper input before a session is started. If the user desires to use MIMO for any state-space related options, he needs the input (A,B,C) system at his disposal. The user will require this (A,B,C) for any state coordinate system option. MIMO does not require the D matrix if one exists for its work.

The user may obtain singular values, singular vectors, and estimate the condition number with respect to inversion without inputting the (A,B,C) system through Option 2 first.

MIMO has various options to offer which are listed and described in detail in the next section. Most options explain themselves. They all ask for the desired inputs. However, if at any time should a question arise, referral to this user's manual should answer it.

TABLE OF CONTENTS

DESCRIPTION OF OPTIONS

OPTION 0 List all options

```
TYPE '0'(ZERO)  FOR A LIST OF OPTIONS
TYPE OPTION NUMBER >0


SIXTEEN OPTIONS ARE NOW PROVIDED
OPTION 0:     LIST OPTIONS
OPTION 1:     STOP
OPTION 2:     OBTAIN THE BALANCED (A',B',C'),GIVEN (A,B,C)
OPTION 3:     OBTAIN THE DISCRETE TIME F AND G MATRICES
              AND THE CORRESPONDING DISCRETE TIME
              CONTROLLABILITY,OBSERVABILITY AND
              HANKEL MATRICES
OPTION 4:     PLOT SINGULAR VALUES OF DISCRETE
              CONTROLLABILITY,OBSERVABILITY,AND
              HANKEL MATRICES VS. SAMPLE TIME
OPTION 5:     OBTAIN THE CONTINUOUS TIME
              CONTROLLABILIY AND OBSERVABILTITY
              MATRICES
OPTION 6:     ESTIMATE CONDITION NUMBER OF A MATRIX
OPTION 7:     OBTAIN THE SINGULAR VALUES FOR A GIVEN
              MATRIX-----NEED NOT BE SQUARE
OPTION 8:     BODE PLOT OF USER SPECIFIED TRANSFER
              FUNCTION(BALANCED SYSTEM) FOR GIVEN
              ORDER SYSTEM
OPTION 9:     DISPOSE PLOTS TO PLOTTER
OPTION 10:    SAVE ALL INFORMATION IN FILE MEMORE
OPTION 11:    RECOVER INFO FROM LOCAL FILE--MEMORE
OPTION 12:    TOTAL INTERFACE
OPTION 13:    OBTAIN OUTPUT NORMAL REP.
              OPTION 2 MUST BE EXECUTED FIRST
OPTION 14:    OBTAIN OUTPUT-PREDICTIVE REP.
OPTION 15:    OBTAIN THE CONTINUOUS TIME CONTROLLABILITY
              AND OBSERVABILITY GRAMMIANS FOR THE INPUT
              (A,B,C) SYSTEM. (WARNING---THE A MATRIX MUST
              HAVE ONLY NEGATIVE EIGENVALUES!!!)
```

OPTION 1 Stop

Terminates execution of MIMO. Saves all current information in local file MEMORE. If OPTION 12 has been executed, the TOTAL interface local file MEMAUX is created.

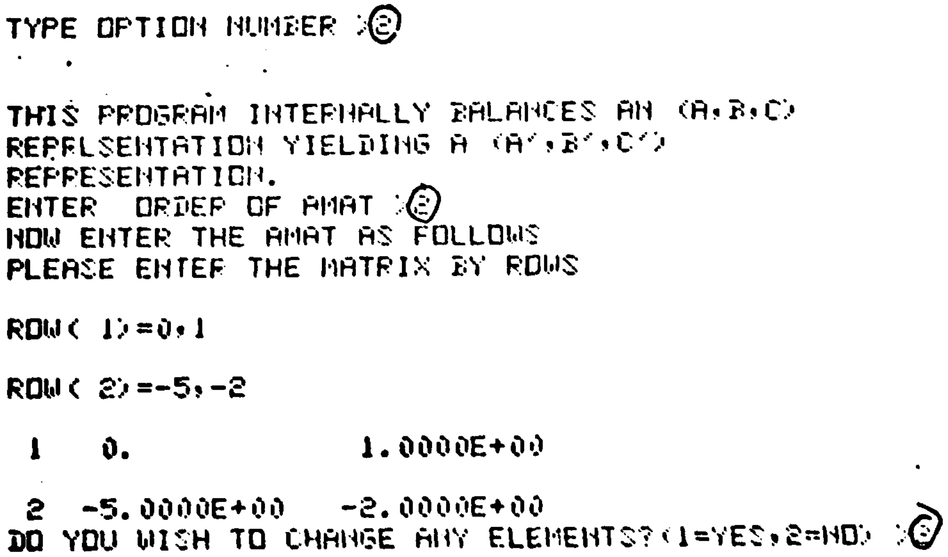
If the user has executed a plotting option, the PLOT file will be created.

This PLOT file will be empty _if_ OPTION 9 has been executed _after_ either of the plotting options(4or8) has been executed. This occurs because of the peculiarities of the software required for OPTION 9. Therefore, the PLOT file may be returned. The PLOT file will contain plots _if_ OPTION 9 has _not_ been executed after either of the plotting options(4or8) has been executed.

OPTION 2 Input (A,B,C) and optionally obtain the internally balanced representation

(The maximum dimension of A,B, or C is 10X10)

a) Options 2b,3,4,5,6,7,8,9,10,11,12,13, and 14 use the (A,B,C) system input by this option. Input must occur by rows. The matrices are printed out for verification, and the opportunity to change mistakes is provided. You may then terminate OPTION 2, or obtain the internally balanced representation.



```
TYPE OPTION NUMBER >2

THIS PROGRAM INTERNALLY BALANCES AN (A,B,C)
REPRESENTATION YIELDING A (A',B',C')
REPRESENTATION.
ENTER  ORDER OF AMAT >2
NOW ENTER THE AMAT AS FOLLOWS
PLEASE ENTER THE MATRIX BY ROWS

ROW( 1)=0,1

ROW( 2)=-5,-2

 1    0.              1.0000E+00

 2  -5.0000E+00   -2.0000E+00
DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) >2
```

```
ENTER THE COLUMN DIMENSION(# OF INPUTS) OF THE BMAT >1
PLEASE ENTER THE MATRIX BY ROWS

ROW( 1)=0

ROW( 2)=1

 1    0.

 2    1.0000E+00
DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) >2
ENTER THE ROW DIMENSION(# OF OUTPUTS) OF THE CMAT
NOTE:  IF SISO SYSTEM, CMAT=(CVEC)TRANSPOSED >1
PLEASE ENTER THE MATRIX BY ROWS

ROW( 1)=1,0

 1    1.0000E+00      0.
DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) >2
```

b) If the balancing path is chosen, you must then choose whether the balanced system should better approximate; (1) the response to an impulse input, or (2) the response to a step input. This is not to say the impulse response balanced system reduced order model will not give excellent results to a step input or any other class of inputs. However, in order to obtain more accurate, lower order models to respond to a step input, the system should be balanced with respect to a step input.

If the step response balancing path is chosen, a D' matrix will be created. MIMO asks the user the desired order of his reduced order model. Once entered, MIMO will obtain the D' matrix which should be added directly to the original D matrix(if one did not exist--this D' matrix becomes the necessary D matrix to provide good results). Note however, this D' matrix is appropriate only for the reduced order model for which it is derived.

Do not attempt to add this D' matrix to any reduced order system other than the order specified by the user. The user should try different combinations of the impulse and step balancing paths. to obtain the most accurate reduced order model for his desired input.

Reduction of order is accomplished by deleting the bottom most states of the resulting balanced representation.

i.e.

$$A_{\text{balanced}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad B_{\text{balanced}} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix} \qquad C_b = [13 \; 14 \; 15]$$

$$A_{\substack{\text{balanced}\\ \text{reduced}}} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \qquad B_{\substack{\text{balanced}\\ \text{reduced}}} = \begin{bmatrix} 10 \\ 11 \end{bmatrix} \qquad C_{b_r} = [13 \; 14]$$

The balancing algorithm requires; (1) the input system to be at least marginally stable, and (2) if the step balancing path is chosen, the input A matrix must *not* be singular.

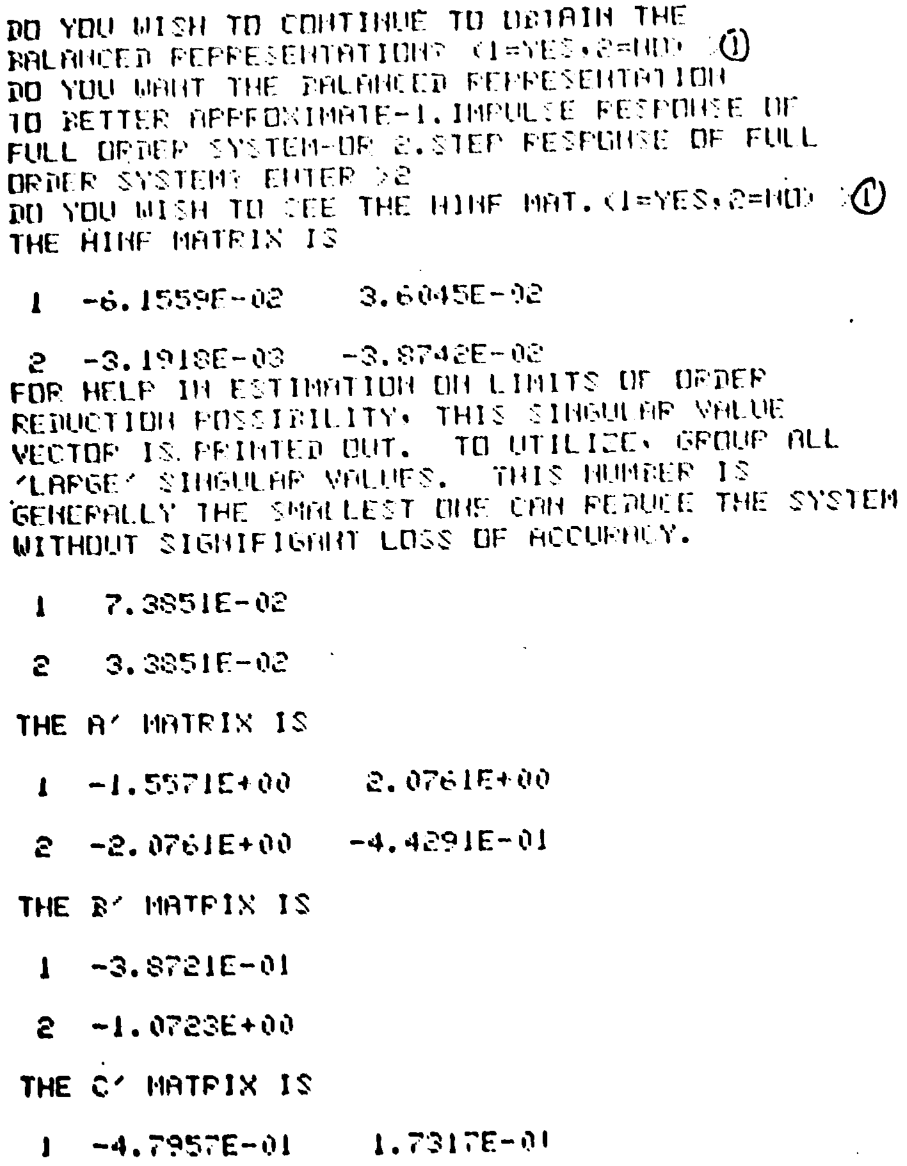
A check for accuracy exists if and only if a SISO(single-input-single-output) system is input. This check consists of an absolute value symmetry occuring in the balanced A matrix. This check does not hold for Mimo(multi-input-multi-output) systems.

The user may print out the H_{INF} matrix used in the balancing algorithm. This matrix's singular values are printed out subsequently. These singular values are related to those of the discrete-time Hankel matrix via the following relation:

$$\lim_{t \to 0} \sigma_{\text{HANKEL}} = \sigma_{H_{INF}}$$; where t is sampling time and σ represents singular values of the infinite order Hankel matrix

The user can group these singular values into "large" and "small" groups. The number of "large" singular values can give an approximation to the lowest possible reduced order model that can be attained without severe loss of accuracy.

The B' and C' for output matrices should be used for time and frequency domain responses. The B' and C' matrices order the inputs and outputs with respect to their l_2 norm. This indicates the "influence" of the inputs, and the "responsiveness" of outputs.



```
DO YOU WISH TO CONTINUE TO OBTAIN THE
BALANCED REPRESENTATION? (1=YES,2=NO) >1
DO YOU WANT THE BALANCED REPRESENTATION
TO BETTER APPROXIMATE-1.IMPULSE RESPONSE OF
FULL ORDER SYSTEM-OR 2.STEP RESPONSE OF FULL
ORDER SYSTEM? ENTER >2
DO YOU WISH TO SEE THE HINF MAT.(1=YES,2=NO) >1
THE HINF MATRIX IS

 1  -6.1559E-02    3.6045E-02

 2  -3.1918E-03   -3.8742E-02
FOR HELP IN ESTIMATION ON LIMITS OF ORDER
REDUCTION POSSIBILITY, THIS SINGULAR VALUE
VECTOR IS PRINTED OUT.  TO UTILIZE, GROUP ALL
'LARGE' SINGULAR VALUES.  THIS NUMBER IS
GENERALLY THE SMALLEST ONE CAN REDUCE THE SYSTEM
WITHOUT SIGNIFIGANT LOSS OF ACCURACY.

 1   7.3851E-02

 2   3.3851E-02

THE A' MATRIX IS

 1  -1.5571E+00    2.0761E+00

 2  -2.0761E+00   -4.4291E-01

THE B' MATRIX IS

 1  -3.8721E-01

 2  -1.0723E+00

THE C' MATRIX IS

 1  -4.7957E-01    1.7317E-01
```

```
FOR OUTPUT THE BMAT IS

 1   3.8721E-01

 2   1.0723E+00
FOR OUTPUT THE CMAT IS

 1   4.7957E-01   -1.7317E-01
ENTER DESIRED ORDER OF REDUCED SYSTEM >1
THE D' MATRIX IF BALANCED SYSTEM IS
REDUCED TO 1 IS .

 1   8.0742E-02
```

<u>OPTION 3</u> <u>Discretize (A,B,C) or (A',B',C') and obtain the F, G, discrete-time controllability, observability, and Hankel matrices</u>

A truncated (50 term) power series expansion is used to obtain the F and G matrices for the (A,B,C) system, or the (A',B',C') system. Both systems are results of OPTION 2. The (F,G,C) matrices are used to obtain the discrete-time controllability, observability, and Hankel matrices.

```
TYPE OPTION NUMBER :3
DO YOU WISH TO SUPPRESS THE PRINTOUTS?(1=YES,2=NO) :2

 DO YOU WISH TO DISCRETIZE THE ORIGINAL(1) SYSTEM
 OR THE BALANCED(2) SYSTEM?  ENTER>1
ENTER THE SAMPLING TIME(T)>.4

THE F MATRIX IS

 1    7.0745E-01     2.4043E-01

 2   -1.2021E+00     2.2659E-01

THE G MATRIX IS

 1    5.8511E-02

 2    2.4043E-01
```

```
THE DISCRETE TIME CONTROLLABILITY
MATRIX IS

 1   5.8511E-02    9.9199E-02

 2   2.4043E-01   -1.5860E-02
THE DISCRETE TIME OBSERVABILITY
MATRIX IS

 1   1.0000E+00    0.

 2   7.0745E-01    2.4043E-01
THE DISCRETE TIME HANKEL MATRIX IS

 1   5.8511E-02    9.9199E-02

 2   9.9199E-02    6.6365E-02
```

OPTION 4 Plot and optionally list singular values and their ratio

The listing may be supressed. The discrete-time controllability, observability, Hankel matrix, or all three matrices' singular values may be plotted and optionally listed. The user may plot any two(2) singular values and their ratio for each matrix versus sample time. The singular values are ordered from minimum to maximum. (Sig(1) is the minimum singular value, Sig(n) the maximum)

The user is asked to input the indices of the singular values he wishes to plot.(i.e. 3,5). If 3,5 is entered, singular value #3, singular value #5, and the ratio singular value #3/singular value #5 is plotted. The first indice entered by the user will subsequently be referred to as the "minimum" singular value. Similarly, the second indice entered by the user will be referred to as "maximum".

USES: The plot of the condition number of the Hankel matrix is a great aid in choice of sample time. The user should pick a sample time that minimizes this condition number. (Condition number= Sig(max)/Sig(min)). This will make the input (F,G,C) system more robust to pertubations in its elements if the sample time is chosen via this process.

The other available plots give the user an idea of the degree of control or the responsiveness of outputs to a given input that a particular system possess.

```
TYPE OPTION NUMBER >4

THE CONTROLLABILITY AND OBSERVABILITY MATRICES EACH
HAVE 2 IMPORTANT SINGULAR VALUES.  THEY ARE ORD-
ERED FROM MINIMUM TO MAXIMUM IN MAGNITUDE. ENTER--
1 TO PLOT CONTROLLABILITY, 2 TO PLOT TO PLOT OBSER-
VABILITY,3 TO PLOT HANKEL, OR 4 TO PLOT ALL >3

YOU MAY PLOT 2 SINGULAR VALUES AND THEIR RATIO VS.
SAMPLE TIME: I.E. SIG(N1),SIG(N2),SIG(N1)/SIG(N2)
ENTER THE 2 SINGULAR VALUE NUMBERS IN DESIRED ORDER
FOR RATIO PLOT--SIG(I.D.1)/SIG(I.D.2)--(I.E. ENTER
2,5--WILL PLOT SINGULAR VALUES #2, AND #5 AND #2/#5)
ENTER >1 2
THE 1ST I.D. WILL BE REFERRED TO AS MIN, 2ND--MAX
DO YOU WISH TO SUPRESS THE PRINTOUT?(1=YES,2=NO) >2

NOTE  :THE FOLLOWING TITLE BLOCKS ARE PROVIDED FOR YOUR
CONVENIENCE.  THE PLOTS ARE ALREADY WELL-LABELED AS TO
WHICH ONES THEY ARE.  THE TITLE BLOCKS MAY BE USED FOR
USER DEFINED LABELS--SUCH AS FIGURE NUMBERS ETC.
PLOT OF THE MAXIMUM SING. VAL. OF HANKEL MAT.
> <--------ENTER TITLE  (50 CHARACTERS MAX)--------> <
>FIGURE 1  MINIMUM SINGULAR VALUE OF HANKEL MAT
  FIGURE 1   MINIMUM SINGULAR VALUE OF HANKEL MAT
PLOT OF THE MINIMUM SING. VAL. OF HANKEL MAT.
> <--------ENTER TITLE  (50 CHARACTERS MAX)--------> <
>FIGURE 2 MAXIMUM SINGULAR VALUE OF HANKEL MAT
  FIGURE 2 MAXIMUM SINGULAR VALUE OF HANKEL MAT
PLOT OF THE CONDITION NUMBER OF HANKEL MAT.
> <--------ENTER TITLE  (50 CHARACTERS MAX)--------> <
>FIGURE 3 RATIO OF SIG1/SIG2
  FIGURE 3 RATIO OF SIG1/SIG2

NOTE: THE CALCULATION DELTA FOR THIS ALGORITHM
IS THE INITIAL TIME. THEREFORE THE INITIAL TIME MUST
```

```
BE LESS THAN ONE. THE FINAL TIME MUST BE WITHIN
98 STEPS OF THE INITIAL TIME
PLEASE ENTER THE BEGINNING SAMPLING TIME
AND THE FINAL SAMPLING TIME >.1,.5
THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME .1 ARE >

 1   2.7990E-03

 2   2.6051E-02
```

OPTION 5 Obtain the continuous time controllability and observability matrices

Obtains the continuous time controllability and observability matrices for (A,B,C) system input by OPTION 2.

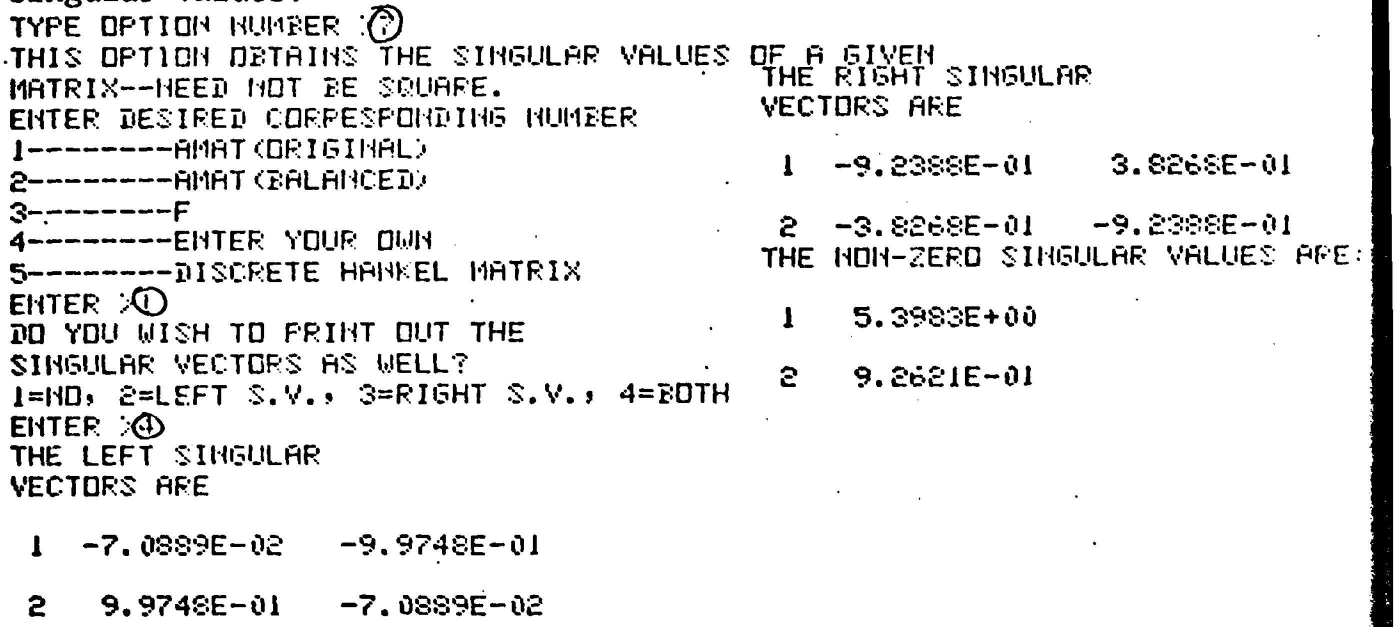
OPTION 6 Estimate condition number of a matrix

This option uses a package of highly efficient LINPACK (SIAM Publishing Co.) subroutines to estimate the condition number of a matrix. This option does not require relatively expensive singular value decompositions. It is useful when the condition number's estimated magnitude is desired. Several options exist. The user can obtain the estimate for A(original), A'(balanced, F(must be created by OPTION 3 first), or the H_{INF} matrix, a matrix input at this time, or the discrete-time Hankel matrix(must be created by OPTION 3 first). Warning---except for the Hankel matrix, all matrices must be square!

```
TYPE OPTION NUMBER >6
THIS OPTION ESTIMATES THE CONDITION NUMBER OF A
SQUARE MATRIX.  ENTER THE NUMBER CORRESPONDING
TO THE DESIRED INPUT MATRIX
1---------AMAT(ORIGINAL)
2---------AMAT(BALANCED)
3---------F
4---------ENTER YOUR OWN
5---------MATRIX TO HELP EVALUATE ORDER REDUCTION
6---------HANKEL MAT.(MUST BE CREATED BY OPT. 3)
ENTER >1

THE ESTIMATED CONDITION NUMBER IS RCOND= 6.25
```

OPTION 7 Prints out singular values and optionally the left and right singular vectors of input matrix

The user can again specify the same matrices as above in OPTION 6, except for the H_{INF} matrix. This option does not require the matrices to be square. The user may print out the left singular vectors, the right singular vectors, both, and the singular values.

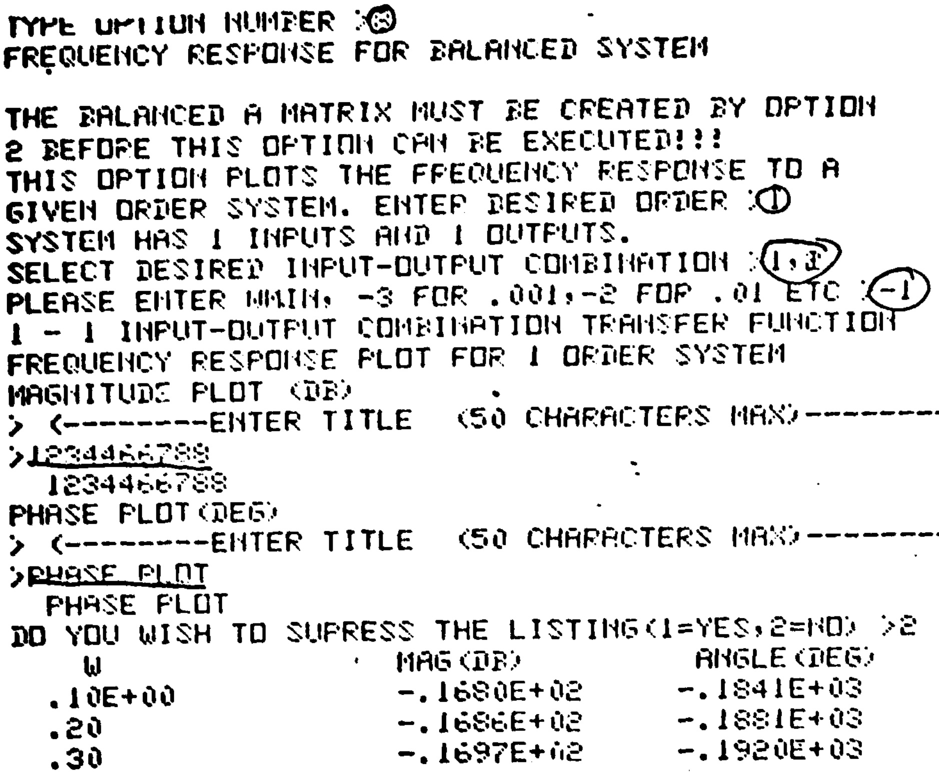


```
TYPE OPTION NUMBER :7
THIS OPTION OBTAINS THE SINGULAR VALUES OF A GIVEN
MATRIX--NEED NOT BE SQUARE.
ENTER DESIRED CORRESPONDING NUMBER
1---------AMAT(ORIGINAL)
2---------AMAT(BALANCED)
3---------F
4---------ENTER YOUR OWN
5---------DISCRETE HANKEL MATRIX
ENTER >1
DO YOU WISH TO PRINT OUT THE
SINGULAR VECTORS AS WELL?
1=NO, 2=LEFT S.V., 3=RIGHT S.V., 4=BOTH
ENTER >4
THE LEFT SINGULAR
VECTORS ARE

 1  -7.0889E-02   -9.9748E-01

 2   9.9748E-01   -7.0889E-02
```

```
THE RIGHT SINGULAR
VECTORS ARE

 1  -9.2388E-01    3.8268E-01

 2  -3.8268E-01   -9.2388E-01
THE NON-ZERO SINGULAR VALUES ARE:

 1   5.3983E+00

 2   9.2621E-01
```

OPTION 8 Frequency response for given order balanced system

This option uses the modal description of a system to calculate the frequency response of a given order balanced system. The magnitude(DB) and phase(DEG) plots will be created. Optionally, a listing of 100 points can be obtained starting with a user specified frequency(RAD/SEC).



```
TYPE OPTION NUMBER >8
FREQUENCY RESPONSE FOR BALANCED SYSTEM

THE BALANCED A MATRIX MUST BE CREATED BY OPTION
2 BEFORE THIS OPTION CAN BE EXECUTED!!!
THIS OPTION PLOTS THE FREQUENCY RESPONSE TO A
GIVEN ORDER SYSTEM. ENTER DESIRED ORDER >1
SYSTEM HAS 1 INPUTS AND 1 OUTPUTS.
SELECT DESIRED INPUT-OUTPUT COMBINATION >1,1
PLEASE ENTER WMIN, -3 FOR .001,-2 FOR .01 ETC >-1
1 - 1 INPUT-OUTPUT COMBINATION TRANSFER FUNCTION
FREQUENCY RESPONSE PLOT FOR 1 ORDER SYSTEM
MAGNITUDE PLOT (DB)
> <--------ENTER TITLE  (50 CHARACTERS MAX)--------> <
>1234466788
  1234466788
PHASE PLOT(DEG)
> <--------ENTER TITLE  (50 CHARACTERS MAX)--------> <
>PHASE PLOT
  PHASE PLOT
DO YOU WISH TO SUPRESS THE LISTING(1=YES,2=NO) >2
    W                   MAG(DB)          ANGLE(DEG)
  .10E+00              -.1680E+02       -.1841E+03
  .20                  -.1686E+02       -.1881E+03
  .30                  -.1697E+02       -.1920E+03
```

OPTION 9 Dispose plots to AFIT plotter

This option requests a queue device for the plot file, closes it, and routes all accumulated plots to the AFIT plotter under your 2 letter user I.D.(when you login-the user I.D. assigned to you--can be obtained by typing "assets"). The plot file is then reopened. This option allows the user to route plots without ceasing execution of MIMO. The plot file must not exist if this program is reexecuted. If this option is not used, the standard operations with the plot file may be performed once execution of MIMO is terminated. See discussion on plot file options after this listing section.

OPTION 10 Update file MEMORE

This option updates the local file MEMORE with current, pertinent information, but does not cause termination of the program(this is automatically executed when OPTION 1 is chosen).

OPTION 11 Recover information from file MEMORE

This option recovers pertinent information from local file MEMORE created by OPTION 1 or OPTION 10. This option coupled with option 10 allows the user to cease execution of MIMO to do other things and come back and recover to his original point.

OPTION 12 TOTAL interface

This option interfaces MIMO to TOTAL. This saves the user from typing in large matrices into TOTAL to obtain responses. The option explains itself. The following list is the key to the interface:

MIMO	TOTAL
AMAT(original)	UMAT
BMAT(original)	VMAT
CMAT(original)	WMAT
AMAT(balanced)	XMAT
BMAT(balanced-output)	YMAT
CMAT(balanced-output)	ZMAT

These matrices are stored in a file called MEMAUX. This file is created by this option and will exist at termination of this program. This option (12) should be executed after the balanced system is obtained for options 13 and 14 will overwrite the original balanced system. The user should note that TOTAL will not alter MIMO in any way. In other words the interface is one-way. No TOTAL obtained matrices may be interfaced back into MIMO.

```
TYPE OPTION NUMBER >12

(A,B,C) AND (A',B',C') HAVE BEEN STORED IN SUCH A WAY
THAT TOTAL CAN DIRECILY ACCESS THEM.
THROUGH USE OF THE COPY COMMAND UNDER TOTAL FULL FLEXI-
BILITY IS EXERCISED. THE FOLLOWING LIST IS THE KEY

          MIMO                         STORED IN TOTAL
          A ORIGINAL                    UMAT
          B ORIGINAL                    VMAT
          C ORIGINAL                    WMAT
          A BALANCED                    XMAT
          B BALANCED-OUTPUT             YMAT
          C BALANCED-OUTPUT             ZMAT

 USING TOTAL'S OPTION 25 THE TRANSFER FUNCTION MAY BE
OBTAINED. WHEN MIMO IS TERMINATED THE ABOVE INFORMATION
IS STORED IN MASS STORAGE FORMAT ON THE LIKEWISE TOTAL
NAMED FILE-----MEMAUX---THIS FILE MUST EXIST PRIOR TO
TRYING TO INTERFACE WITH TOTAL!!!!!!!!
THE FOLLOWING PROCEDURE SHOULD BE UTILIZED.
FIRST COPY (UMAT,VMAT,WMAT) OR (XMAT,YMAT,ZMAT) INTO
(AMAT,BMAT,CMAT).  THEN TYPE OPTION 18 IN TOTAL.
WHEN ASKED FOR THE NUMBER OF STATES--ENTER THE FULL
NUMBER IF USING THE ORIGINAL SYSTEM.  ENTER THE DESIRED
ORDER IF USING THE BALANCE SYSTEM. THEN FOR EACH
ELEMENT, TYPE A *(ASTERIK) WHICH MEANS TOTAL WILL KEEP
THAT LOCATION THE SAME. THEN ENTER NO FOR THE DMAT AND
KMAT IF APPROPRIATE. THEN OPTION 25 CAN BE EXECUTED.
```

*****EXAMPLE OF INTERFACE WITH TOTAL AND THE USE OF TOTAL*****

```
TYPE OPTION NUMBER >1
ALL DATA UPDATED IN LOCAL FILE MEMORE
     STOP  END OF EXECUTION--ALL PLOT(S)
 IN FILE PLOT
        .589 CP SECONDS EXECUTION TIME
FILE QUOTA EXCEEDED
..RETURN,LGO,MEMORE,PLOT
..TOTAL
WELCOME TO TOTAL--VERSION 1.4
TYPE HELP FOR INTRO, TYPE 99 FOR NEW FEATURES BULLETIN

OPTION > 'UMAT,VMAT,WMAT,XMAT,YMAT,ZMAT

  COL >   1            2             3
ROW
 1      0.          1.000         0.
 2      0.          0.            1.000
 3     -1.000      -2.000        -2.000


  COL >   1            2
ROW
 1      1.000       0.
 2      0.          1.000
 3      2.000       3.500
```

```
  COL >    1           2          3
ROW
 1      1.000       2.500      1.000


  COL >    1           2          3
ROW
 1      -.1195      -.1735     -.9220
 2      -.1492      -1.171     -.3290
 3       .9315       .3384     -.7098


  COL >    1           2
ROW
 1       .1584       .8306
 2       .4128       .4956
 3       1.217       2.514


  COL >    1           2          3
ROW
 1      -.8456      -.6450      2.793


OPTION >  COPY,UMAT,AMAT,COPY,VMAT,BMAT,COPY,GMAT,CMAT
COPY COMPLETE
COPY COMPLETE
COPY COMPLETE

OPTION > 18
IS THE SYSTEM (1) CONTINUOUS OR (2) DISCRETE? >   1

THE EQUATIONS YOU ARE ABOUT TO INPUT HAVE THE FORM:

                 XDOT(T) = [ AMAT ]X(T) + [ BMAT ]U(T)
                    Y(T) = [ CMAT ]X(T) + [ DMAT ]U(T)
   WHERE            U(T) =  GAIN*( R(T) - [ KMAT ]X(T) )

AND   X  IS A VECTOR OF  N  STATE VARIABLES
      U  IS A VECTOR OF  M  INPUTS
      Y  IS A VECTOR OF  L  OUTPUTS

ENTER NO. OF STATES,INPUTS,OUTPUTS >  3,2,1
ENTER AMAT WITH 3 ROWS AND 3 COLUMNS.
ENTER 3 ELEMENTS PER ROW:
ROW 1 > ◆,◆,◆
ROW 2 > ◆,◆,◆,
ROW 3 > ◆,◆,◆

  COL >    1              2             3
ROW
 1       0.           1.000         0.
 2       0.           0.            1.000
 3     -1.000        -2.000        -2.000


ENTER BMAT WITH 3 ROWS AND 2 COLUMNS.

ENTER 3 ELEMENTS PER COLUMN:
COLUMN 1 > ◆,◆,◆
COLUMN 2 > ◆,◆,◆

  COL.>    1              2
ROW
 1       1.000        0.
 2       0.           1.000
 3       2.000        3.500


ENTER CMAT WITH 1 ROWS AND 3 COLUMNS.

ENTER 3 ELEMENTS PER ROW:
ROW 1 > ◆,◆,◆

  COL >    1              2             3
ROW
 1       1.000        2.500         1.000


IS THERE A DIRECT-TRANSMISSION (D MATRIX)--YES OR NO? > NO
DMAT SET TO 1 BY 2 ZERO MATRIX (OPTION 78)
IS THERE A STATE-VARIABLE FEEDBACK MATRIX--YES OR NO? > NO
KMAT HAS BEEN SET TO 2 BY 3 ZERO MATRIX
THE STATE-SPACE REPRESENTATION IS COMPLETE.

OPTION > 25
SYSTEM HAS 2 INPUT(S) AND 1 OUTPUT(S).
SELECT WHICH TRANSFER FUNCTION IS DESIRED:
ENTER WHICH INPUT, WHICH OUTPUT > 1,1
GTF(S) AND HTF(S) CALCULATED FROM STATE-SPACE
TYPE:   GTF      OR      HTF       FOR RESULTS
```

```
OPTION >  GTF,HTF
                        FORWARD-LOOP TRANSFER FUNCTION


                    GK= ( GNK/GDK )=    3.000

                          GTF(S) NUMERATOR
 I             GNPOLY(I)                          GZERO(I)
 1      (    3.000      )S** 2        (  -.2929     ) + J(    0.          )
 2      (    6.000      )S** 1        (  -1.707     ) + J(    0.          )
 3      (    1.500      )                       GNK=    3.000

                          GTF(S) DENOMINATOR
 I             GDPOLY(I)                          GPOLE(I)
 1      (    1.000      )S** 3        (  -.5000     ) + J(    .8660       )
 2      (    2.000      )S** 2        (  -.5000     ) + J(  -.8660        )
 3      (    2.000      )S** 1        (  -1.000     ) + J(    0.          )
 4      (    1.000      )                       GDK=    1.000

                        FEEDBACK-LOOP TRANSFER FUNCTION


                    HK= ( HNK/HDK )=    0.

                          HTF(S) NUMERATOR
 I             HNPOLY(I)                          HZERO(I)
 1      (    0.         )                       HNK=    0.

                          HTF(S) DENOMINATOR
 I             HDPOLY(I)                          HPOLE(I)
 1      (    3.000      )S** 2        (  -.2929     ) + J(    0.          )
 2      (    6.000      )S** 1        (  -1.707     ) + J(    0.          )
 3      (    1.500      )                       HDK=    3.000

OPTION >  COPY,GTF,OLTF
COPY COMPLETE

OPTION >  CLOSED,OFF,39
OPEN-LOOP MODE SET

TYPES OF INPUT:  (1) IMPULSE, (2) STEP, (3) RAMP,
(4) PULSE, (5) SINE...SELECT ONE > 2
ENTER STEP INPUT MAGNITUDE >.1


OPTION >  34
   CONTINUOUS TIME RESPONSE FOR OLTF(S)
   WITH STEP INPUT OF STRENGTH = .1

ENTER INITIAL TIME, FINAL TIME > 0,10
> [--------ENTER TITLE  (50 CHARACTERS MAX)--------] <
> TIME RESPONSE OF 3RD ORDER ORIGINAL SYSTEM
  TIME RESPONSE OF 3RD ORDER ORIGINAL SYSTEM
```

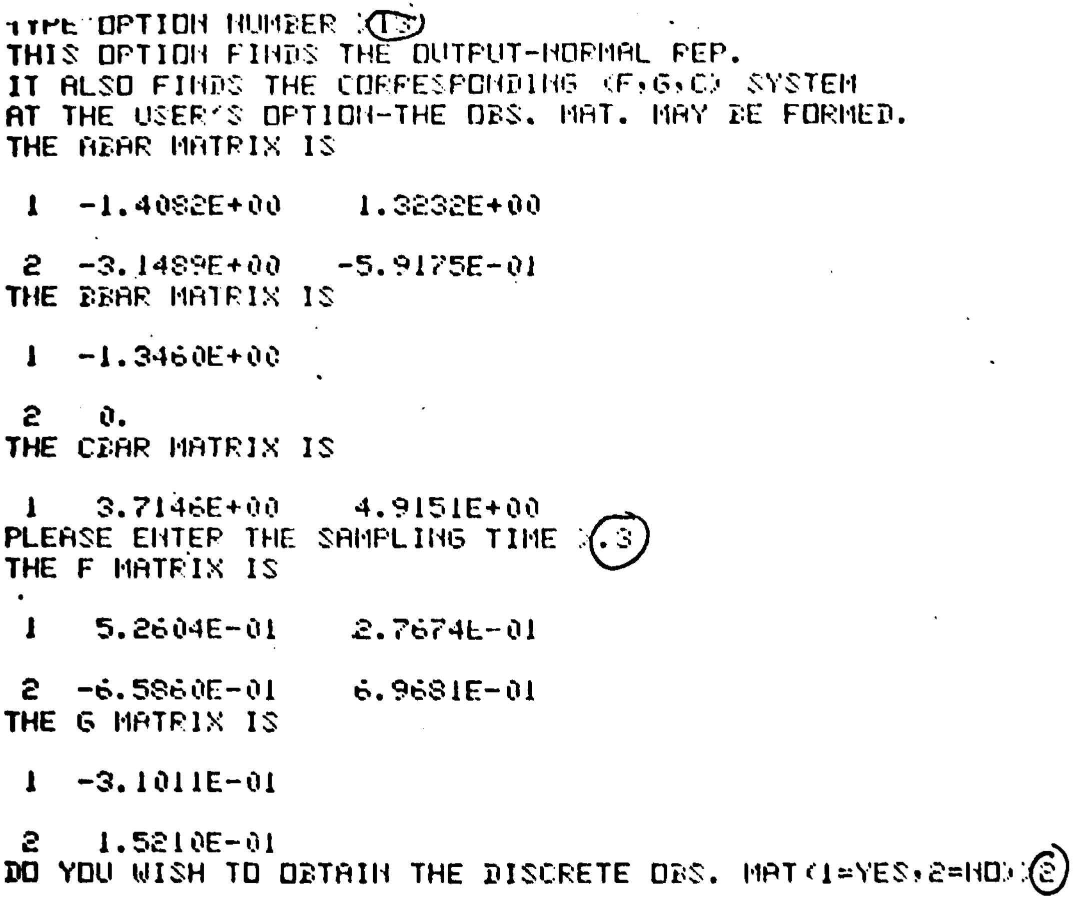
******EXAMPLE OF REDUCING THE ORDER OF THE BALANCED SYSTEM****

```
OPTION >  COPY,XMAT,AMAT,COPY,YMAT,BMAT,COPY,ZMAT,CMAT
COPY COMPLETE
COPY COMPLETE
COPY COMPLETE

OPTION > 18
IS THE SYSTEM (1) CONTINUOUS OR (2) DISCRETE? > 1

THE EQUATIONS YOU ARE ABOUT TO INPUT HAVE THE FORM:

                XDOT(T) = [ AMAT ]X(T) + [ BMAT ]U(T)
                   Y(T) = [ CMAT ]X(T) + [ DMAT ]U(T)
   WHERE           U(T) =  GAIN*( R(T) - [ KMAT ]X(T) )

AND    X  IS A VECTOR OF  N  STATE VARIABLES
       U  IS A VECTOR OF  M  INPUTS
       Y  IS A VECTOR OF  L  OUTPUTS

ENTER NO. OF STATES,INPUTS,OUTPUTS >  2,2,1
ENTER AMAT WITH 2 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:
ROW 1 > *,*
ROW 2 > *,*

  COL >    1            2
ROW
 1      -.1195       -.1735
 2      -.1492       -1.171


ENTER BMAT WITH 2 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:
ROW 1 > *,*
ROW 2 > *,*

  COL >    1            2
ROW
 1       .1584        .8306
 2       .4128        .4956


ENTER CMAT WITH 1 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:
ROW 1 > *,*

  COL >    1            2
ROW
 1      -.8456       -.6450
```

```
IS THERE A DIRECT-TRANSMISSION (D MATRIX)--YES OR NO? > NO
DMAT SET TO 1 BY 2 ZERO MATRIX (OPTION 78)
IS THERE A STATE-VARIABLE FEEDBACK MATRIX--YES OR NO? > NO
KMAT HAS BEEN SET TO 2 BY 2 ZERO MATRIX
THE STATE-SPACE REPRESENTATION IS COMPLETE.

OPTION > 25
SYSTEM HAS 2 INPUT(S) AND 1 OUTPUT(S).
SELECT WHICH TRANSFER FUNCTION IS DESIRED:
ENTER WHICH INPUT, WHICH OUTPUT > 1,1
GTF(S) AND HTF(S) CALCULATED FROM STATE-SPACE
TYPE:   GTF      OR      HTF       FOR RESULTS

OPTION > GTF
                     FORWARD-LOOP TRANSFER FUNCTION


                       GK= ( GNK/GDK )=   -.4002

                          GTF(S) NUMERATOR
  I         GNPOLY(I)                           GZERO(I)
  1      (  -.4002     )S** 1        (  -.2819     ) + J(    0.        )
  2      (  -.1128     )                       GNK=    -.4002

                         GTF(S) DENOMINATOR
  I         GDPOLY(I)                           GPOLE(I)
  1      (   1.000     )S** 2        (  -.9545E-01) + J(    0.        )
  2      (   1.290     )S** 1        (  -1.195     ) + J(    0.        )
  3      (   .1140     )                       GDK=     1.000

OPTION >  STOP
LOCAL FILE--PLOT--CONTAINS CALCOMP PLOT(S)
ALL INFO IN TOTAL HAS BEEN SAVED IN LOCAL FILE--MEMORY.
     STOP
      1.408 CP SECONDS EXECUTION TIME
FILE QUOTA EXCEEDED
..FILES
--LOCAL FILES--
 $INPUT     $OUTPUT     MIMO       *PERM      $INPUT
 $OUTPUT    *JIMLIB    *IMSL       *TOTAL      MEMAUX
  PLOT       MEMORY
FILE QUOTA EXCEEDED
```

OPTION 13 Output normal state coordinate system

This option uses the (A',B',C') system obtained from Option 2 to result in a new (A*,B*,C*) system which is balanced with respect to observability conditions only.



```
TYPE OPTION NUMBER :13
THIS OPTION FINDS THE OUTPUT-NORMAL REP.
IT ALSO FINDS THE CORRESPONDING (F,G,C) SYSTEM
AT THE USER'S OPTION-THE OBS. MAT. MAY BE FORMED.
THE ABAR MATRIX IS

 1  -1.4082E+00    1.3232E+00

 2  -3.1489E+00   -5.9175E-01
THE BBAR MATRIX IS

 1  -1.3460E+00

 2   0.
THE CBAR MATRIX IS

 1   3.7146E+00    4.9151E+00
PLEASE ENTER THE SAMPLING TIME :.3
THE F MATRIX IS

 1   5.2604E-01    2.7674E-01

 2  -6.5860E-01    6.9681E-01
THE G MATRIX IS

 1  -3.1011E-01

 2   1.5210E-01
DO YOU WISH TO OBTAIN THE DISCRETE OBS. MAT(1=YES,2=NO):2
```

OPTION 14 Output predictive state coordinate system

This option creates an (A*,B*,C*) system such that the resulting observability matrix is indeed the identity matrix!

```
TYPE OPTION NUMBER >14
THIS OPTION FINDS THE OUTPUT PREDICTIVE REP.
ENTER SAMPLING TIME >.34
THE F′ MATRIX IS

 1   4.8453E-01    3.7659E-01

 2   3.7659E-01    2.9270E-01
THE G′ MATRIX IS

 1   4.4554E-02

 2   3.4629E-02
THE C′ MATRIX IS

 1   6.2341E-01    4.8453E-01
TYPE OPTION NUMBER >1
```

For more information on this system the user should refer to Major J. Gary Reid's EE 5.10 class notes.

OPTION 15 Obtain the controllability and observability grammians

This option obtains the controllability and observability grammians for the (A,B,C) sytem input in OPTION 2. WARNING----The A matrix must contain only negative eigenvalues.

```
TYPE OPTION NUMBER >15
THE CONTROLLABILITY GRAMMIAN IS

 1   4.9999E-02    6.9574E-16

 2   6.9574E-16    2.5000E-01

THE OBSERVABILITY GRAMMIAN IS

 1   4.4999E-01    9.9998E-02

 2   9.9998E-02    4.9999E-02
```

PLOT FILE OPTIONS

A. OPTION 9 (this program)

1) Execute MIMO---create plots
2) Use OPTION 9---automatically routes plots to the AFIT terminal under your 2 letter user I.D. (see explanation of OPTION 9).

B. Standard Route (use if after C., or if desired route to other building besides AFIT (building 640--BB))

1) Execute MIMO---create plots
2) Terminate MIMO (OPTION 1) <u>without</u> executing OPTION 9 for the plots to be routed
3) Type: Route Plot, TID=BB, ST=CSB, DC=PT (where BB could be any other terminal I.D.)

C. To use CCPREV to preview plots (WARNING--user must be at a TEKTRONICS terminal)

1) Execute MIMO---create plots
2) Terminate MIMO without executing OPTION 9 for the plots to be previewed
3) Type:
```
ATTACH CCPREV, ID=AFIT
CCPREV, PLOT
WINDOW,0,0,50,50,N,R,E,P,999
```
4) Plots will be small. However, if all plots are not on screen use the MOVE command, i.e. Type:
```
MOVE,delta-x,delta-y
```
where delta-x, and delta-y are in screen inches. For example, to move the entire string of plots to the left 2 screen inches, and up 3 screen inches, Type;
```
MOVE,-2,3.
```
5) Then Type:
```
R,E,P,999
```
This will replot the string of plots. Repeat steps 4 and 5 until plot string is on the screen (if impossible refer to CCPREV users manual).
6) Type:
```
SAVE,1
```
7) Type:
```
CR
```
This will display a vertical, and a horizontal line on the screen. Using the white rotating dials located on the terminal, the lines should be moved to the bottom left hand corner of the plot to be previewed. Type any letter on the keyboard (if the user is on the 1200 Baud 4014 terminal you must also type a carriage return!). The lines will momentarily disappear, and then reappear. Move the lines to the upper, right hand corner of the plot to be previewed. Again, type any character on the keyboard. This defines a new window.

8) Type:

   ```
   N,R,E,P,999
   ```

   This will display the delineated plot over the entire screen.
9) After examination of this plot, and if another of the plot string is to be previewed, Type:

   ```
   USE,1
   ```
10) Now, repeat steps 7-9 for a different plot.
11) When you are finished, Type:

   ```
   END
   ```
12) You may then return the plot file, or execute sequences B or E.

E. To catalog the plot file as a permanent file

1) Execute MIMO
2) Terminate MIMO without executing OPTION 9 for the string of plots desired to be stored.
3) Type:

   ```
   REQUEST,NAME,*PF
   ```

   (where NAME is any character string but "plot" or a current local file name)
4) Type:

   ```
   COPY,PLOT,NAME
   CATALOG,NAME,PLOTFIL
   ```

   (where "plotfil" is any name you desire)
5) This is useful is you are not on a TEKTRONICS terminal at the time of execution of MIMO, and later desire to preview those plots with CCPREV. (if that is the case--you merely attach "procfil" as a local file later and then Type:

   ```
   CCPREV,XXXXX
   ```

   (whereXXXXXis the local file name you attached "procfil"under)

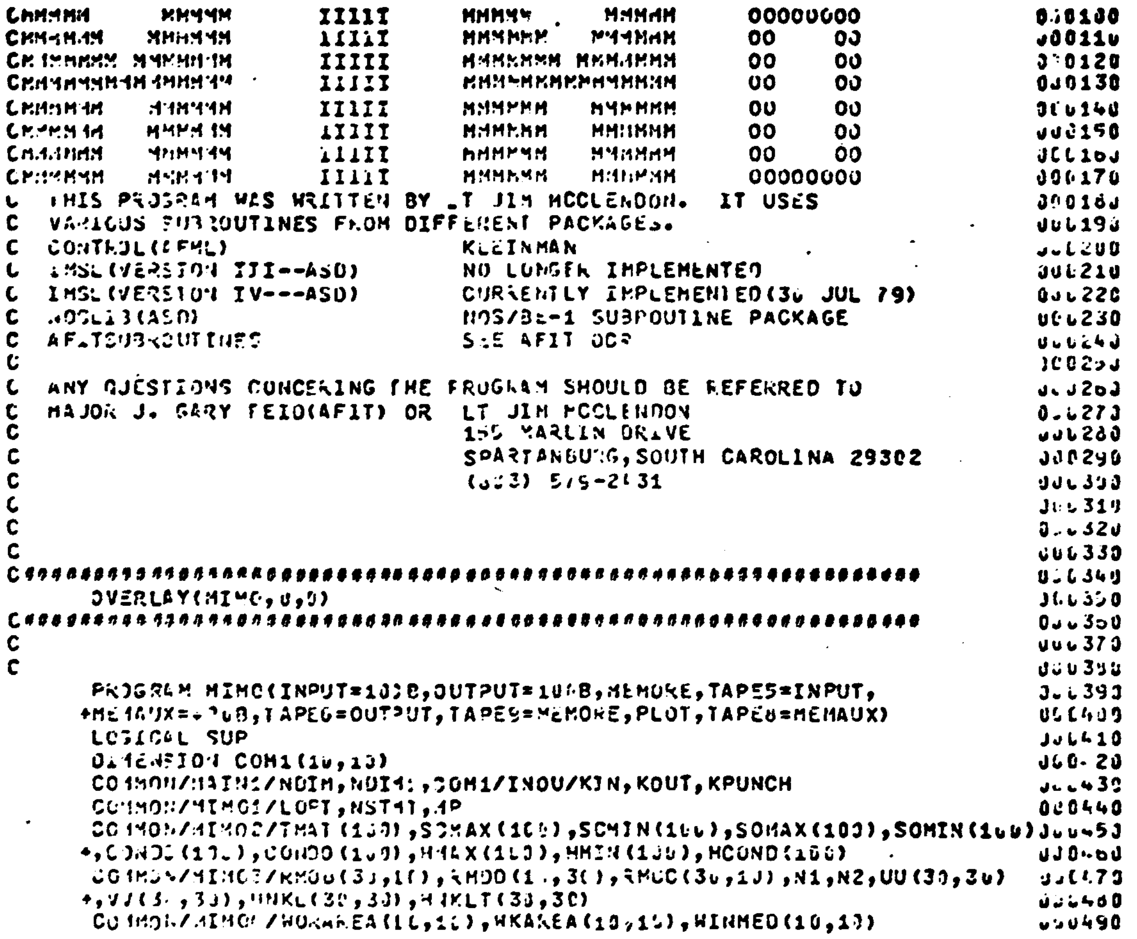
APPENDIX D

LISTING OF MIMO

August 1979

LISTING OF MIMO

This Appendix contains the source listing for MIMO. The sequence numbers are on the listing to assist the user in using the program.



```
CMMMMM     MMMMM        IIIII     MMMMM     MMMMM      00000000             000100
CMMMMMM   MMMMMM        IIIII     MMMMMM   MMMMMM      00    00             000110
CMMMMMMM MMMMMMM        IIIII     MMMMMMM MMMMMMM      00    00             000120
CMMMMMMMMMMMMMMM        IIIII     MMMMMMMMMMMMMMM      00    00             000130
CMMMMMM   MMMMMM        IIIII     MMMMMM   MMMMMM      00    00             000140
CMMMMMM   MMMMMM        IIIII     MMMMMM   MMMMMM      00    00             000150
CMMMMMM   MMMMMM        IIIII     MMMMMM   MMMMMM      00    00             000160
CMMMMMM   MMMMMM        IIIII     MMMMMM   MMMMMM      00000000             000170
C  THIS PROGRAM WAS WRITTEN BY LT JIM MCCLENDON.  IT USES                   000180
C  VARIOUS SUBROUTINES FROM DIFFERENT PACKAGES.                             000190
C  CONTROL(AFML)                   KLEINMAN                                 000200
C  IMSL(VERSION III--ASD)          NO LONGER IMPLEMENTED                    000210
C  IMSL(VERSION IV---ASD)          CURRENTLY IMPLEMENTED(30 JUL 79)         000220
C  MOSLIB(ASD)                     NOS/BE-1 SUBROUTINE PACKAGE              000230
C  AFITSUBROUTINES                 SEE AFIT OCP                             000240
C                                                                           000250
C  ANY QUESTIONS CONCERNING THE PROGRAM SHOULD BE REFERRED TO               000260
C  MAJOR J. GARY FEID(AFIT) OR  LT JIM MCCLENDON                            000270
C                               155 MARLIN DRIVE                            000280
C                               SPARTANBURG,SOUTH CAROLINA 29302            000290
C                               (803) 579-2131                              000300
C                                                                           000310
C                                                                           000320
C                                                                           000330
C########################################################################   000340
      OVERLAY(MIMO,0,0)                                                     000350
C########################################################################   000360
C                                                                           000370
C                                                                           000380
      PROGRAM MIMO(INPUT=100B,OUTPUT=100B,MEMORE,TAPE5=INPUT,               000390
     +MEMAUX=400B,TAPE6=OUTPUT,TAPE9=MEMORE,PLOT,TAPE8=MEMAUX)              000400
      LOGICAL SUP                                                           000410
      DIMENSION COM1(10,10)                                                 000420
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                     000430
      COMMON/MIMO1/LOPT,NSTAT,MP                                            000440
      COMMON/MIMO2/TMAT(100),SOMAX(100),SCMIN(100),SOMAX(100),SOMIN(100)000450
     +,CONDC(100),CONDO(100),HMAX(100),HMIN(100),HCOND(100)                 000460
      COMMON/MIMO3/RMOB(30,10),RMOD(10,30),RMOC(30,10),N1,N2,UU(30,30)      000470
     +,VV(30,30),HNKL(30,30),HNKLT(30,30)                                   000480
      COMMON/MIMO4/WORKAREA(10,10),WKAREA(10,10),WINMED(10,10)              000490
      COMMON/MIMO7/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10
     +),MC,NC
      COMMON/MIMO6/SIGC(10,10),SIGO(10,10),UC(10,10),UCTRAN(10,10),VOUT(
     +10,10)
      COMMON/MIMO7/BBAR(10,10),SIGO2IN(10,10),Z(10,10),CBAR(10,10),TINV(
     +10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      COMMON/MIMO8/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     +VHTRAN(10,10),SIGC2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      COMMON/MIMO9/UO(10,10),UOTRAN(10,10),SIG2O(10,10)
      COMMON/MIMO10/VIN(10,10),SIG2IN(10,10),VINTRAN(10,10),WINSQ(10,10)
     +,WOUTSQ(10,10),VH(10,10)
      COMMON/MIMO11/ATRAN(10,10),BTRAN(10,10),CTRAN(10,10),WSQUAR(10,10)
     +,WSQUAR2(10,10),BPROD(10,10),CPROD(10,10)
      COMMON/PLOT1/I,SIVS(30,1),SUF,IWAS,JWAS,IWAS1,JWAS1
      COMMON/SAVE1/NITST
      COMMON/MASSTO/INMASS(17)
      COMMON/DIS/NDSP
      NDSP=1
      PRINT*,"THE 'PLOT' FILE MUST NOT EXIST IF AND WHEN"
      PRINT*,"THIS PROGRAM IS REEXECUTED!!!!!"
      DO 9606 IO=1,10
      DO 9606 JO=1,10
      DO 9606 KO=1,30
      A(IO,JO)=0.
      B(IO,JO)=0.
      RMOD(IO,KO)=0.
      F(IO,JO)=0.
      G(IO,JO)=0.
      C(IO,JO)=0.
      BBAR(IO,JO)=0.
      CBAR(IO,JO)=0.
      AREAL(IO,JO)=0.
      HINF(IO,JO)=0.
      BREAL(IO,JO)=0.
      CREAL(IO,JO)=0.
 9606 CONTINUE
      DO 9607 IO=1,30
      DO 9607 JO=1,30
      DO 9607 KO=1,10
      RMOB(IO,KO)=0.
      RMCC(IO,KO)=0.
      SIVS(IO,1)=0.
      MNKL(IO,JO)=0.
9665  CONTINUE
      DO 9666 IO=1,100
      TMAT(IO)=0.
      HMIN(IO)=0.
      HCOND(IO)=0.
      HMAX(IO)=0.
      SCMAX(IO)=0.
      SOMAX(IO)=0.
      SOMIN(IO)=0.
      CONDC(IO)=0.
      CONDO(IO)=0.
      SCMIN(IO)=0.
9666  CONTINUE
      N1=N2=N=M=MC=NC=MB=NB=0
      MF=0
      NDIM=10
      NDIM1=11
      KIN=5
      KOUT=6
      KPUNCH=7
      SUP=.FALSE.
      CALL DATE(AAA)
      CALL CLOCK(AA)
      WRITE(6,100) AAA,AA
100   FORMAT(1X,"MIMO IS EXECUTING ON ",A10," AT " ,A10)
C
C
      PRINT',"  "
      PRINT',"WELCOME TO MIMO--------VERSION 1"
      PRINT',"IF ANY OPTION UTILIZING (A,B,C) IS DESIRED;INPUT MUST"
      PRINT',"OCCUR THROUGH OPTION 2.  ALL OTHER OPTIONS USE THE INPUT"
      PRINT',"MATRICES FOR THEIR WORK. MATRIX SIZES ARE: A,B,C--10X10"
      PRINT',"THE CONTROLLABILTIY AND OBSERVABILITY OPTIONS CAN HANDLE"
      PRINT',"3 INPUTS AND 3 OUTPUTS IF A 10X10 SYSTEM IS INPUT.  IN"
      PRINT',"OTHER WORDS THE CONTROLLABILITY MATRIX IS LIMITED TO A"
      PRINT',"10X30, WHILE THE OBSERVABILITY MATRIX IS LIMITED TO A"
      PRINT',"30X10 MATRIX."
      PRINT*," "
      PRINT*," "
      PRINT*,"TYPE '0'(ZERO)  FOR A LIST OF OPTIONS "
5     CALL OVERLAY(4HMIMO,1,0)
      MP=MP+1
      IF(LOPT.EQ.1)CALL OVERLAY(4HMIMO,10,0)
6     IF(LOPT.EQ.1)STOP "END OF EXECUTION--ALL PLOT(S) IN FILE PLOT"
      IF(NSTMT.EQ.1)CALL OVERLAY(4HMIMO,2,0)
      IF(NSTMT.EQ.200)CALL OVERLAY(4HMIMO,3,0)
      IF(NSTMT.EQ.200)CALL OVERLAY(4HMIMO,12,0)
      IF(NSTMT.EQ.250)CALL OVERLAY(4HMIMO,4,0)
      IF(NSTMT.EQ.5)CALL OVERLAY(4HMIMO,5,0)
      IF(NSTMT.EQ.6)CALL OVERLAY(4HMIMO,6,0)
      IF(NSTMT.EQ.7)CALL OVERLAY(4HMIMO,7,0)
      IF(NSTMT.EQ.6)CALL OVERLAY(4HMIMO,17,0)
      IF(NSTMT.EQ.8)CALL OVERLAY(4HMIMO,8,0)
      IF(NSTMT.EQ.9)CALL OVERLAY(4HMIMO,9,0)
      IF(NSTMT.EQ.10)CALL OVERLAY(4HMIMO,10,0)
      IF(NSTMT.EQ.11)CALL OVERLAY(4HMIMO,10,0)
      IF(NSTMT.EQ.12)CALL OVERLAY(4HMIMO,11,0)
      IF(NSTMT.EQ.13)CALL OVERLAY(4HMIMO,13,0)
      IF(NSTMT.EQ.14)CALL OVERLAY(4HMIMO,14,0)
      IF(NSTMT.EQ.15)CALL OVERLAY(4HMIMO,15,0)
      GO TO 5
      END
C*******************************************************
      OVERLAY(1,0)
C*******************************************************
      PROGRAM BALANCE
      COMMON/MIMO1/LOPT,NSTMT,MP
      IF(MP.GE.1)GO TO 10
5     PRINT*,"SIXTEEN OPTIONS ARE NOW PROVIDED"
      PRINT*,"OPTION 0:   LIST OPTIONS"
      PRINT*,"OPTION 1:   STOP"
      PRINT*,"OPTION 2:   OBTAIN THE BALANCED (A',B',C'),GIVEN (A,B,C)"
      PRINT*,"OPTION 3:   OBTAIN THE DISCRETE TIME F AND G MATRICES"
      PRINT*,"            AND THE CORRESPONDING DISCRETE TIME "
      PRINT*,"            CONTROLLABILITY,OBSERVABILITY AND"
      PRINT*,"            HANKEL MATRICES"
      PRINT*,"OPTION 4:   PLOT SINGULAR VALUES OF DISCRETE"
      PRINT*,"                     CONTROLLABILITY,OBSERVABILITY,AND"
      PRINT*,"                     HANKEL MATRICES VS. SAMPLE TIME"
      PRINT*,"OPTION 5:            OBTAIN THE CONTINUOUS TIME"
      PRINT*,"                     CONTROLLABILIY AND OBSERVABILTITY"
      PRINT*,"                     MATRICES"
      PRINT*,"OPTION 5:            ESTIMATE CONDITION NUMBER OF A MATRIX"
      PRINT*,"OPTION 7:            OBTAIN THE SINGULAR VALUES FOR A GIVEN"
      PRINT*,"                     MATRIX-----NEED NOT BE SQUARE"
      PRINT*,"OPTION 8:            BODE PLOT OF USER SPECIFIED TRANSFER"
      PRINT*,"                     FUNCTION(BALANCED SYSTEM) FOR GIVEN"
      PRINT*,"                     ORDER SYSTEM"
      PRINT*,"OPTION 9:            DISPOSE PLOTS TO PLOTTER"
      PRINT*,"OPTION 10:           SAVE ALL INFORMATION IN FILE MEMORE"
      PRINT*,"OPTION 11:           RECOVER INFO FROM LOCAL FILE--MEMORE"
      PRINT*,"OPTION 12:           TOTAL INTERFACE"
      PRINT*,"OPTION 13:           OBTAIN OUTPUT NORMAL REP."
      PRINT*,"                     OPTION 2 MUST BE EXECUTED FIRST"
      PRINT*,"OPTION 14:           OBTAIN OUTPUT-PREDICTIVE REP."
      PRINT*,"OPTION 15:           OBTAIN THE CONTINUOUS TIME CONTROLLABILITY"
      PRINT*,"                     AND OBSERVABILITY GRAMMIANS FOR THE INPUT "
      PRINT*,"                     (A,B,C) SYSTEM. (WARNING---THE A MATRIX MUST"
      PRINT*,"                     HAVE ONLY NEGATIVE EIGENVALUES!!!)"
      PRINT*," "
10    PRINT*,"TYPE OPTION NUMBER > "
      READ*,LOPT
      IF(LOPT.EQ.1)GO TO 5
      IF(LOPT.EQ.2)NSTMT=1
      IF(LOPT.EQ.3)NSTMT=200
      IF(LOPT.EQ.4)NSTMT=250
      IF(LOPT.EQ.5)NSTMT=5
      IF(LOPT.EQ.6)NSTMT=6
      IF(LOPT.EQ.7)NSTMT=7
      IF(LOPT.EQ.8)NSTMT=8
      IF(LOPT.EQ.9)NSTMT=9
      IF(LOPT.EQ.10)NSTMT=10
      IF(LOPT.EQ.11)NSTMT=11
      IF(LOPT.EQ.12)NSTMT=12
      IF(LOPT.EQ.13)NSTMT=13
      IF(LOPT.EQ.14)NSTMT=14
      IF(LOPT.EQ.15)NSTMT=15
      END
C************************************************************
      OVERLAY(2,0)
C************************************************************
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C+THIS ALGORITHM WAS DEVELOPED BY B.C. MOORE OF THE UNIVERSITY +
C+OF TORONTO.  AN ASYMPTOTICALLY STABLE SYSTEM MUST BE INPUT   +
C+THIS MODEL REDUCTION TECHNIQUE PRESERVES THE TRANSIENT WELL. +
C+THE USER SHOULD NOTE HOWEVER THERE MAY BE A STEADY-STATE OFF-+
C+SET PRESENT IN THE REDUCED ORDER MODEL.  IN OTHER WORDS THE  +
C+IMPULSE RESPONSE FUNCTION IS PRESERVED WHILE THE TRANSFER    +
C+FUNCTION MAY HAVE A OFFSET IN THE REDUCTION PROCESS          +
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      PROGRAM ABC
      COMMON/MATN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH
      COMMON/MIMO /WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO1/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10)
     +,C(10,10),MC,NC
      COMMON/MIMO2/SIGC(10,10),SIGO(10,10),UC(10,10),UCTRAN(10,10),
     +VOUT(10,10)
      COMMON/MIMO3/BBAR(10,10),SIGO2IN(10,10),T(10,10),CBAR(10,10),
     +TINV(10,10),AREAL(10,10),PREAL(10,10),CREAL(10,10)
      COMMON/MIMO4/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     +VHTRAN(10,10),SIGC2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      COMMON/MIMO5/UO(10,10),UOTRAN(10,10),SIG2O(10,10)
      COMMON/MIMO10/VIN(10,10),SIG2IN(10,10),VINTRAN(10,10),WINSQ(10,10)
     +,WOUTSQ(10,10),VH(10,10)
      COMMON/MIMO11/ATRAN(10,10),BTRAN(10,10),CTRAN(10,10)
     +,WSQUAR(10,10),WSQUAR2(10,10),BPROD(10,10),CPROD(10,10)
5     PRINT ,"THIS PROGRAM INTERNALLY BALANCES AN (A,B,C)"
      PRINT ,"REPRESENTATION YIELDING A (A',B',C')"
      PRINT ,"REPRESENTATION."
1     PRINT*,"ENTER  ORDER OF AMAT > "
      READ*,M
      IF(M.GT.10)PRINT*,"SIZE LIMITED TO 10TH ORDER"
      IF(M.GT.10)GO TO 1
      IF(EOF(5LINPUT).NE.0)STOP "END OF SESSION"
      N=MB=NC=M
```

```
C     READ MATRICES FROM USER
      PRINT*,"NOW ENTER THE AMAT AS FOLLOWS"
      CALL READM(M,N,A)
C      ALLOW USER TO CHANGE ELEMENTS IF DESIRED
      CALL PRINTM(M,N,A)
      CALL MISTAKE(M,N,A)
      PRINT*,"ENTER THE COLUMN DIMENSION(# OF INPUTS) OF THE BMAT > "
      READ*,NB
      CALL READM(MB,NB,B)
      CALL PRINTM(MB,NB,B)
      CALL MISTAKE(MB,NB,B)
      PRINT*,"ENTER THE ROW DIMENSION(# OF OUTPUTS) OF THE CMAT"
      PRINT*,"NOTE:  IF SISO SYSTEM, CMAT=(CVEC)TRANSPOSED > "
      READ*,MC
      CALL READM(MC,NC,C)
      CALL PRINTM(MC,NC,C)
      CALL MISTAKE(MC,NC,C)
      PRINT*,"DO YOU WISH TO CONTINUE TO OBTAIN THE"
      PRINT*,"BALANCED REPRESENTATION? (1=YES,2=NO) > "
      READ*,IAH
      IF(IAH.EQ.2)GO TO 9999
      PRINT*,"DO YOU WANT THE BALANCED REPRESENTATION"
      PRINT*,"TO BETTER APPROXIMATE-1.IMPULSE RESPONSE OF"
      PRINT*,"FULL ORDER SYSTEM-OR 2.STEP RESPONSE OF FULL"
      PRINT*,"ORDER SYSTEM? ENTER > "
      READ*,IAM
      IF(IAM.EQ.1)GO TO 3
      DO 4I=1,N
      DO 4J=1,N
      WKAREA(I,J)=A(I,J)
4     CONTINUE
      CALL LINV1F(WKAREA,N,10,WORKAREA,1,WINMEO,IER)
      CALL MMULT(WORKAREA,B,BBAR,N,MB,NB)
      DO 6I=1,MB
      DO 6J=1,NB
      BTRAN(J,I)=BBAR(I,J)
6     CONTINUE
      GO TO 999
3     DO 10I=1,MB,1
      DO 10J=1,NB,1
      BTRAN(J,I)=B(I,J)                                                  00002900
10    CONTINUE                                                           00002910
999   CONTINUE                                                           00002920
      DO 20 I=1,M,1                                                      00002930
      DO 20 J=1,N,1                                                      00002940
      ATRAN(J,I)=A(I,J)                                                  00002950
20    CONTINUE                                                           00002960
      DO 30 I=1,MC,1                                                     00002970
      DO 30 J=1,NC,1                                                     00002980
      CTRAN(J,I)=C(I,J)                                                  00002990
30    CONTINUE                                                           00003000
      IF(IAM.EQ.1)CALL MMULT(B,BTRAN,BPROD,MB,NB,MB)                     00003010
      IF(IAM.EQ.2)CALL MMULT(BBAR,BTRAN,BPROD,MB,NB,MB)                  00003020
      CALL MLINEQ(N,ATRAN,BPROD,WSQUAR,.01)                              00003030
      CALL MMULT(CTRAN,C,CPROD,NC,MC,NC)                                 00003040
      CALL MLINEQ(NC,A,CPROD,WSQUAR2,.01)                                00003050
      CALL MMULT(WSQUAR,CTRAN,WINMED,NC,NC,MC)                           00003060
      CALL MMULT(C,WINMED,WOUTSQ,MC,NC,MC)                               00003070
      CALL MMULT(WSQUAR2,B,WINMED,MB,MB,NB)                              00003080
      CALL MMULT(BTRAN,WINMED,WINSQ,NB,MB,NB)                            00003090
      CALL LSVALF(WOUTSQ,MC,MC,10,10,1,WKAREA,SIG2O,VOUT,VOUTRAN)        00003100
      CALL LSVALF(WINSQ,NB,NB,10,10,1,WKAREA,SIG2IN,VIN,VINTRAN)         00003110
      CALL MMULT(B,VIN,BBAR,MB,NB,NB)                                    00003120
      DO 35 I=1,MC,1                                                     00003130
      DO 35 J=1,MC,1                                                     00003140
      VOUTRAN(J,I)=VOUT(I,J)                                             00003150
35    CONTINUE                                                           00003160
      DO 40 IT=1,NB                                                      00003170
      DO 40 JT=1,NB                                                      00003180
      VINTRAN(JT,IT)=VIN(IT,JT)                                          00003190
40    CONTINUE                                                           00003200
      CALL MMULT(VOUTRAN,C,CBAR,MC,MC,NC)                                00003210
      CALL LSVALF(WSQUAR,N,N,10,10,1,WKAREA,SIGC,UC,UCTRAN)              00003220
      CALL LSVALF(WSQUAR2,N,N,10,10,1,WKAREA,SIGO,UO,UOTRAN)             00003230
      DO 50 I=1,N,1                                                      00003240
      DO 50 J=1,N,1                                                      00003250
      UCTRAN(J,I)=UC(I,J)                                                00003260
      UOTRAN(J,I)=UO(I,J)                                                00003270
50    CONTINUE                                                           00003280
      DO 55 I=1,N,1                                                      00003290
      SIGO(I,1)=SQRT(SIGO(I,1))
      SIGC(I,1)=SQRT(SIGC(I,1))
 60   CONTINUE
      DO 70I=1,N,1
      DO 70J=1,N,1
      SIGO2(I,J)=0.000
      SIGC2(I,J)=0.000
      SIGO2(I,I)=SIGO(I,1)
      SIGC2(I,I)=SIGC(I,1)
 70   CONTINUE
      CALL MMULT(SIGO2,UOTRAN,HINMED,N,N,N)
      CALL MMULT(HINMED,UC,HKAREA,N,N,N)
      CALL MMULT(HKAREA,SIGC2,HINF,N,N,N)
      PRINT*,"DO YOU WISH TO SEE THE HINF MAT.(1=YES,2=NO) > "
      READ*,IYES
      IF(IYES.EQ.1)PRINT*,"THE HINF MATRIX IS "
      IF(IYES.EQ.1)CALL PRINTM(N,N,HINF)
      CALL LSVALR(HINF,N,N,10,10,1,HKAREA,SIGH,UH,VH)
      DO 75I=1,N,1
      DO 75J=1,N,1
      VHTRAN(J,I)=VH(I,J)
 75   CONTINUE
      DO 80I=1,N,1
      HKAREA(I,1)=SIGH(I,1)
      SIGH(I,1)=SQRT(SIGH(I,1))
 80   CONTINUE
      DO 90I=1,N,1
      DO 90J=1,N,1
      SIGH2(I,J)=0.000
      SIGH2(I,I)=SIGH(I,1)
 90   CONTINUE
C     THESE SINGULAR VALUES ARE THE LIMITS OF THE DISCRETE TIME EXTENDED
C     INFINITE DIMENSIONAL HANKEL MATRIX
      PRINT*,"FOR HELP IN ESTIMATION ON LIMITS OF ORDER"
      PRINT*,"REDUCTION POSSIBILITY, THIS SINGULAR VALUE"
      PRINT*,"VECTOR IS PRINTED OUT.  TO UTILIZE, GROUP ALL"
      PRINT*,"'LARGE' SINGULAR VALUES.  THIS NUMBER IS"
      PRINT*,"GENERALLY THE SMALLEST ONE CAN REDUCE THE SYSTEM"
      PRINT*,"WITHOUT SIGNIFICANT LOSS OF ACCURACY."
      CALL PRINTM(N,1,HKAREA)
      CALL LINV1F(SIGO2,N,1J,SIGO2IN,1,WKAREA,IER)
      CALL MMULT(MO,SIGO2IN,WKAREA,N,N,N)
      CALL MMULT(WKAREA,UH,WINMED,N,N,N)
      CALL MMULT(VINMED,SIG42,WKAREA,N,N,N)
      CALL MMULT(CBAR,WKAREA,CREAL,MC,NC,NC)
      C LL MMULT(A,WKAREA,WINMED,N,N,N)
      CALL LINV1F(WKAREA,N,1I,TINV,1,WORAREA,IER)
      CALL MMULT(TINV,WINMED,AREAL,N,N,N)
      CALL MMULT(TINV,BBAR,BREAL,N,MC,NC)
      PRINT*," "
      PRINT*,"THE A' MATRIX IS"
      CALL PRINTM(N,N,AREAL)
      PRINT*," "
      PRINT*,"THE B' MATRIX IS"
      CALL PRINTM(MB,NB,BREAL)
      PRINT*," "
      PRINT*,"THE C' MATRIX IS"
      CALL PRINTM(MC,NC,CREAL)
      MF=MP+1
      PRINT*," "
      CALL MMULT(BREAL,VINTRAN,BBAR,MB,NB,NB)
      PRINT*,"FOR OUTPUT THE BMAT IS "
      CALL PRINTM(MB,NB,BBAR)
      CALL MMULT(VOUT,CREAL,CBAR,MC,MC,NC)
      PRINT*,"FOR OUTPUT THE CMAT IS "
      CALL PRINTM(MC,NC,CBAR)
      IF(IOM.EQ.1)GO TO 9993
      DO 91I=1,M
      DO 91J=1,N
      WKAREA(I,J)=A(I,J)
91    CONTINUE
      CALL LINV1F(WKAREA,N,16,WORAREA,1,WINMED,IER)
      CALL MMULT(C,WOKAREA,WINMED,MC,NC,N)
      CALL MMULT(WINMED,B,WKAREA,MC,NC,NB)
      PRINT*,"ENTER DESIRED ORDER OF REDUCED SYSTEM > "
      READ*,IORDER
      IF(IORDER.GT.N)PRINT*,"ILLEGAL SYSTEM ORDER"
      IF(IORDER.GT.N)GO TO 9993
      DO 92I=1,IORDER
      DO 92J=1,IORDER
      WINMED(I,J)=AREAL(I,J)
92    CONTINUE
      CALL LINV1F(WINMED,IORDER,10,WORAREA,1,TINV,IER)
      CALL MMULT(CBAR,WORAREA,WINMED,MC,IORDER,IORDER)
      CALL MMULT(WINMED,BBAR,WORAREA,MC,IORDER,NB)
      DO 93I=1,MC
      DO 93J=1,NB
      WINMED(I,J)=-(WKAREA(I,J)-WORAREA(I,J))
93    CONTINUE
      PRINT*,"THE D* MATRIX IF BALANCED SYSTEM IS "
      PRINT ,"REDUCED TO ",IORDER," IS ."
      CALL PRINTM(MC,NB,WINMED)
9999  CONTINUE
      END
C*****************************************************************
C     THIS SUBROUTINE READS IN A GIVEN MATRIX
C*****************************************************************
      SUBROUTINE READM(M,N,RRMAT)
      DIMENSION RRMAT(10,10)
      PRINT*,"PLEASE ENTER THE MATRIX BY ROWS"
100   DO 105I=1,M,1
10    WRITE(6,15) I
15    FORMAT(1H ,"ROW(",I2,")=")
      READ*,(RRMAT(I,J),J=1,N,1)
105   CONTINUE
110   RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE ALLOWS USER T  CHANGE ERRONEOUS DATA
C*****************************************************************
      SUBROUTINE MISTAKE(M,N,RMAT)
      DIMENSION RMAT(10,10)
      PRINT*,"DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) > "
      READ*,INDIC
      MIST=2
      IF(INDIC.EQ.2)GO TO 2
3     PRINT*,"TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
      PRINT*,"THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
      PRINT*,"MATRIX ELEMENT SEPARATED BY COMMAS > "
      MIST=1
      READ*,MM,NN,CORREC
      RMAT(MM,NN)=CORREC
      PRINT*,"ANY MORE? (1=YES,2=NO)"
      READ*,INOU
      IF(INOU.EQ.2)GO TO 2
      GO TO 3
2     IF(MIST.EQ.1)CALL PRINTM(M,N,RMAT)
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C*************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20I=1,M,1
      WRITE (6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C*************************************************************
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.0D0
      DO 1J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
C#################################################################
      OVERLAY(3,0)
C#################################################################
C     THIS PROGRAM DISCRETIZES THE (A,B,C) SYSTEM AND PROVIDES THE
```

```
C     DISCRETE-TIME F,G,CONTROLLABILITY,OBSERVABILITY, AND HANKEL
C     MATRICES FOR A GIVEN SAMPLE TIME.
C     (USES A TRUNCATED(50 TERM) POWER SERIES APPROXIMATION)
C
C
      PROGRAM SINGVAL
      LOGICAL SUPP
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH
      COMMON/MIMO2/RMOB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,
     +UU(30,30),VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO6/WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO6/A(10,10),M,N,B(10,10),MB,NB,F(10,10),
     +G(10,10),C(10,10),MC,NC
      COMMON/MIMO7/BBAR(10,10),SIGO2IN(10,10),Z(10,10),CBAR(10,10),TINV(
     +10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      SUPP=.FALSE.
      PRINT*,"DO YOU WISH TO SUPRESS THE PRINTOUTS?(1=YES,2=NO) > "
      READ*,ISIT
      IF(ISIT.EQ.1)SUPP=.TRUE.
      PRINT*," "
      PRINT*," DO YOU WISH TO DISCRETIZE THE ORIGINAL(1) SYSTEM"
      PRINT*," OR THE BALANCED(2) SYSTEM?  ENTER> "
      READ*,IPASS
      PRINT*,"ENTER THE SAMPLING TIME(T)> "
      READ*,T
      IF(IPASS.EQ.1)CALL DKREET(F,G,N,MB,NB,WKAREA,A,B,T)
      IF(IPASS.EQ.2)CALL DKREET(F,G,N,MB,NB,WKAREA,AREAL,BBAR,T)
      IF(SUPP)GO TO 100
      PRINT*," "
      PRINT*,"THE F MATRIX IS"
      CALL PRINTM(N,N,F)
      PRINT*," "
      PRINT*,"THE G MATRIX IS"
      CALL PRINTM(N,NB,G)
C      THIS PART OF THE PROGRAM FINDS THE DISCRETE TIME
C      CONTROLLABILITY MATRIX.
C
100   CONTINUE
      CALL CONBTY(F,G,MB,NB,N,RMOD,WORAREA,WINMED,N1)
      IF(SUPP)GO TO 200
```

```
150   PRINT*,"THE DISCRETE TIME CONTROLLABILITY"
      PRINT*,"MATRIX IS "
      CALL PRINTM(N,N1,RMOD)
C     THIS PART OF THE PROGRAM FINDS THE DISCRETE TIME
C     OBSERVABILITY MATRIX
C
200   CONTINUE
      IF(IPASS.EQ.1)CALL OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)
      IF(IPASS.EQ.2)CALL OBS(MC,NC,N,F,WINMED,WORAREA,RMOD,CBAR,N2)
      IF(SUPP)GO TO 300
250   PRINT*,"THE DISCRETE TIME OBSERVABILITY "
      PRINT ,"MATRIX IS "
      CALL PRINTM1(N2,N,RMOB)
C     THIS PART OF THE PROGRAM FINDS THE
C     DISCRETE HANKEL MATRIX
C
300   CONTINUE
      CALL VMULFF(RMOB,RMOD,N2,N,N1,30,10,HNKL,30,IER)
      IF(SUPP)GO TO 400
      PRINT*,"THE DISCRETE TIME HANKEL MATRIX IS "
      CALL PRINTMI(N2,N1,HNKL)
400   CONTINUE
      END
C*****************************************************************
C     THIS SUBROUTINE MULTIPLIES TWO MATIICES AND STORES IN C
C*****************************************************************
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.0D0
      DO 1J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
C*****************************************************************
```

```
C     THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C*****************************************************************
      SUBROUTINE PRINTM1(M,N,EMAT)
      DIMENSION EMAT(30,10)
      DO 20 I=1,M,1
      WRITE (6,10) I,(EMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE FINDS EXP(AMAT*T) AND ITS INTEGRAL
C     VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C*****************************************************************
      SUBROUTINE DISKRET(N,A,DEL,EA,EAINT,NT)
      DIMENSION EA(10,10),EAINT(10,10),A(10,10),Z(10,10),
     CWINMED(10,10),CBAR(10,10)
      DO 100 I=1,N
      DO 100 J=1,N
      EAINT(I,J)=0.0D0
      EAINT(I,I)=DEL
      Z(I,J)=EAINT(I,J)
100   CONTINUE
      DO 400 K=2,NT
      DO 150 I=1,N
      DO 150 J=1,N
      CBAR(I,J)=A(I,J)*(DEL/FLOAT(K))
150   CONTINUE
      CALL MMUL(Z,CBAR,N,N,N,WINMED)
      DO 160 I=1,N
      DO 160 J=1,N
      Z(I,J)=WINMED(I,J)
      EAINT(I,J)=EAINT(I,J)+WINMED(I,J)
160   CONTINUE
400   CONTINUE
      CALL MMUL(A,EAINT,N,N,N,Z)
      DO 500 I=1,N
      DO 500 J=1,N
      EA(I,J)=0.0D0
      EA(I,I)=1.0D0
```

```
640   CONTINUE
      DO 660 I=1,N
      DO 660 J=1,N
      EA(I,J)=EA(I,J)+Z(I,J)
660   CONTINUE
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE ACCEPTS EXP(AMAT*T) AND ITS INTEGRAL
C     AND FORMS THE DISCRETE F MATRIX AND G MATRIX
C*************************************************************
      SUBROUTINE DKREET(F,G,N,MB,N3,WKAREA,A,B,T)
      DIMENSION F(10,10),G(10,10),WKAREA(10,10),A(10,10)
      DIMENSION B(10,10)
      CALL DISKRET(N,A,T,F,WKAREA,50)
      CALL MMULT(WKAREA,B,G,N,MB,N3)
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C     MATRIX
C*************************************************************
      SUBROUTINE CONBTY(F,G,MB,NB,N,RMOD,WORAREA,WINMED,N1)
      DIMENSION F(10,10),G(10,10),RMOD(10,30),WINMED(10,10)
      DIMENSION WORAREA(10,10)
      DO 100 I=1,MB
      DO 100 J=1,NB
      RMOD(I,J)=G(I,J)
100   CONTINUE
      N1=NB
      IF(N.EQ.1)GO TO 150
      CALL MMULT(F,G,WINMED,N,N,NB)
      DO 120 I=1,MB
      DO 120 J=1,NB
      K=J
      N1=NB
      RMOD(I,N1+J)=WINMED(I,J)
120   CONTINUE
      N1=N1+K
      IF(N.EQ.2)GO TO 150
      NN=N-1                                                              006500
      DO 140 L=2,NN,1                                                     006510
      CALL MMULT(F,WINMED,WORAREA,N,N,NB)                                 006520
      DO 130 I=1,MP,1                                                     006530
      DO 130 J=1,NB,1                                                     006540
      KMOD(I,N1+J)=WORAREA(I,J)                                           006550
      WINMED(I,J)=WORAREA(I,J)                                            006560
      K=J                                                                 006570
  130 CONTINUE                                                            006580
      N1=N1+K                                                             006590
  140 CONTINUE                                                            006600
  150 RETURN                                                              006610
      END                                                                 006620
C*********************************************************************    006630
C     THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX             006640
C*********************************************************************    006650
      SUBROUTINE OBS(MC,NC,N,F,WINMED,WORAREA,KMOD,C,N2)                  006660
      DIMENSION WINMED(10,10),WORAREA(10,10),C(10,10)                     006670
      DIMENSION F(10,10),RMOB(30,10)                                      006680
      DO 200 I=1,MC                                                       006690
      DO 200 J=1,NC                                                       006700
      RMOB(I,J)=C(I,J)                                                    006710
      N2=MC                                                               006720
  200 CONTINUE                                                            006730
      IF(N.EQ.1)GO TO 250                                                 006740
      CALL MMULT(C,F,WINMED,MC,NC,N)                                      006750
      DO 220 I=1,MC                                                       006760
      DO 220 J=1,NC                                                       006770
      K=I                                                                 006780
      N2=MC                                                               006790
      RMOB(N2+I,J)=WINMED(I,J)                                            006800
  220 CONTINUE                                                            006810
      N2=N2+K                                                             006820
      IF(N.EQ.2)GO TO 250                                                 006830
      NN=N-1                                                              006840
      DO 240 L=2,NN,1                                                     006850
      CALL MMULT(WINMED,F,WORAREA,MC,NC,N)                                006860
      DO 230 I=1,MC                                                       006870
      DO 230 J=1,NC                                                       006880
      RMOB(N2+I,J)=WORAREA(I,J)                                           006890
      WINMED(I,J)=WORAREA(I,J)
      K=I
230   CONTINUE
      N2=N2+K
240   CONTINUE
250   RETURN
      END
C***********************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C***********************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20 I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C***********************************************************************
C     THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX
C***********************************************************************
      SUBROUTINE PRINTM2(M,N,DMAT)
      DIMENSION DMAT(30,30)
      DO 20 I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C#######################################################################
      OVERLAY(4,0)
C#######################################################################
C     THIS PROGRAM PLOTS AS WELL AS LISTS THE DISCRETE-TIME
C     CONTROLLABILITY, OBSERVABILITY AND HANKEL MATRICES SINGULAR
C     VALUES AS A FUNCTION OF SAMPLING TIME.  THIS IS USEFUL TO THE
C     USER AS ANOTHER CONSTRAINT ON THE CHOICE OF APPROPRIATE SAMPLE
C     TIME FOR THE SYSTEM IN CONSIDERATION
C
C
      PROGRAM PLOTSIV
      LOGICAL SUP
      DIMENSION ID1(17),ID2(17),ID3(17),ID4(17),ID5(17),ID6(17)
     +,ID7(17),ID8(17),ID9(17)
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH
      COMMON/MIMO2/IMAT(100),SCMAX(100),SCMIN(100),SOMAX(100),
     +SOMIN(100),CONDC(100),CONDO(100),HMAX(100),HMIN(100),HCOND(100)
      COMMON/MIMO3/RMOB(30,10),RMOD(10,30),RMOC(30,10),
     +N1,N2,UU(30,30),VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO4/WOFAREA(10,10),WMAREA(10,10),WINMED(10,10)
      COMMON/MIMO5/A(10,10),M,N,B(10,10),MB,NB,F(10,10),
     +G(10,10),C(10,10),MC,NC
      COMMON/MIMO6/SIGC(10,10),SIGU(10,10),JC(10,10),
     +JCTRAN(10,10),VOUT(10,10)
      COMMON/MIMO7/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     +VHTRAN(10,10),SIGC2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      COMMON/PLOT1/I/1,SIVS(11,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      DATA ID1(1)/20HMIN. SINGULAR VALUE /
      DATA ID1(3)/20H  VS. SAMPLE TIME   /
      DATA ID1(5)/20HCONTROLLABILITY MAT./
      DATA ID1(7)/20H     PROGRAM MIMO   /
      DATA ID1(9)/20HSAMPLE TIME(SECS)   /
      DATA ID1(11)/20H  MIN.SIG. VALUE    /
      DATA ID2(1)/20HMAX. SINGULAR VALUE /
      DATA ID2(3)/20H VS. SAMPLE TIME    /
      DATA ID2(5)/20HCONTROLLABILITY MAT./
      DATA ID2(7)/20H      PROGRAM MIMO  /
      DATA ID2(9)/20H SAMPLE TIME(SECS)  /
      DATA ID2(11)/20H MAX. SING. VALUE   /
      DATA ID3(1)/20HMIN. SINGULAR VALUE /
      DATA ID3(3)/20H VS. SAMPLE TIME    /
      DATA ID3(5)/20HOBSERVABILITY MATRIX/
      DATA ID3(7)/20H    PROGRAM MIMO    /
      DATA ID3(9)/20H SAMPLE TIME(SECS)  /
      DATA ID3(11)/20HMIN. SINGULAR VALUE /
      DATA ID4(1)/20HMAX. SINGULAR VALUE /
      DATA ID4(3)/20H VS. SAMPLE TIME    /
      DATA ID4(5)/20HOBSERVABILITY MATRIX/
      DATA ID4(7)/20H    PROGRAM MIMO    /
      DATA ID4(9)/20H SAMPLE TIME(SECS)  /
      DATA ID4(11)/20HMAX. SINGULAR VALUE /
      DATA ID5(1)/20H  CONDITION NUMBER  /                              007700
      DATA ID5(3)/20H VS. SIMPLE TIME    /                              007710
      DATA ID5(5)/20HCONTRO_LABILITY MAT./                              007720
      DATA ID5(7)/20H   PROGRAM MIMO     /                              007730
      DATA ID5(9)/20H SAMPLE TIME(SECS)  /                              007740
      DATA ID5(11)/20H CONDITION NUMBER   /                             007750
      DATA ID6(1)/20H  CONDITION NUMBER  /                              007760
      DATA ID6(3)/20H VS. SIMPLE TIME    /                              007770
      DATA ID6(5)/20HOBSERVABILITY MATRIX/                              007780
      DATA ID6(7)/20H   PROGRAM MIMO     /                              007790
      DATA ID6(9)/20H SAMPLE TIME(SECS)  /                              007800
      DATA ID6(11)/20H CONDITION NUMBER   /                             007810
      DATA ID7(1)/20HMAX SINGULAR VALUE  /                              007820
      DATA ID7(3)/20H VS. SIMPLE TIME    /                              007830
      DATA ID7(5)/20H  HANKEL MATRIX     /                              007840
      DATA ID7(7)/20H  PROGRAM MIMO      /                              007850
      DATA ID7(9)/20H SAMPLE TIME(SECS)  /                              007860
      DATA ID7(11)/20HMAX SINGULAR VALUE  /                             007870
      DATA ID8(1)/20HMIN SINGULAR VALUE  /                              007880
      DATA ID8(3)/20H VS. SIMPLE TIME    /                              007890
      DATA ID8(5)/20H  HANKEL MATRIX     /                              007900
      DATA ID8(7)/20H  PROGRAM MIMO      /                              007910
      DATA ID8(9)/20H SAMPLE TIME(SECS)  /                              007920
      DATA ID8(11)/20HMIN SINGULAR VALUE  /                             007930
      DATA ID9(1)/20H  CONDITION NUMBER  /                              007940
      DATA ID9(3)/20H VS. SIMPLE TIME    /                              007950
      DATA ID9(5)/20H  HANKEL MATRIX     /                              007960
      DATA ID9(7)/20H  PROGRAM MIMO      /                              007970
      DATA ID9(9)/20H SAMPLE TIME(SECS)  /                              007980
      DATA ID9(11)/20H  CONDITION NUMBER  /                             007990
      SUP=.FALSE.                                                       008000
      PRINT*,"THE CONTROLLABILITY AND OPSERVABILITY MATRICES EACH"      008010
      PRINT*,"HAVE ",N," IMPORTANT SINGULAR VALUES.  THEY ARE ORD-"     008020
      PRINT*,"ERED FROM MINIMUM TO MAXIMUM IN MAGNITUDE. ENTER--"       008030
      PRINT*,"1 TO PLOT CONTROLLABILITY, 2 TO PLOT TO PLOT OBSER-"      008040
      PRINT*,"VABILITY,3 TO PLOT HANKEL, OR 4 TO PLOT ALL > "           008050
      READ*,ILIKE                                                       008060
      PRINT*," "                                                        008070
10    PRINT*,"YOU MAY PLOT 2 SINGULAR VALUES AND THEIR RATIO VS."       008080
      PRINT*,"SAMPLE TIME. I.E. SIG(N1),SIG(N2),SIG(N1)/SIG(N2)"        008090
      PRINT*,"ENTER THE 2 SINGULAR VALUE NUMBERS IN DESIRED ORDER"
      PRINT*,"FOR RATIO PLOT--SIG(I.D.1)/SIG(I.D.2)--(I.E. ENTER"
      PRINT*,"2,5--WILL PLOT SINGULAR VALUES #2, AND #5 AND #2/#5)"
      PRINT*,"ENTER > "
      READ*,IWAS,JWAS
      IF(IWAS.GT.N.OR.JWAS.GT.N)PRINT ,"ILLEGAL ONLY ",N," IMPORTANT"
      IF(IWAS.GT.N.OR.JWAS.GT.N)PRINT*,"SINGULAR VALUES!!!!"
      IF(IWAS.GT.N.OR.JWAS.GT.N) GO TO 10
      PRINT*,"THE 1ST ID WILL BE REFERRED TO AS MIN, 2ND-MAX, RATIO-
     +CONDITION NUMBER"
      PRINT ,"DO YOU WISH TO SUPRESS THE PRINTOUT?(1=YES,2=NO) > "
      READ*,ISIT
      IF(ISIT.EQ.1)SUP=.TRUE.
      PRINT ," "
      PRINT*,"NOTE  :THE FOLLOWING TITLE BLOCKS ARE PROVIDED FOR YOUR"
      PRINT*,"CONVENIENCE.  THE PLOTS ARE ALREADY WELL-LABELED AS TO"
      PRINT*,"WHICH ONES THEY ARE.  THE TITLE BLOCKS MAY BE USED FOR"
      PRINT*,"USER DEFINED LABELS--SUCH AS FIGURE NUMBERS ETC."
      IF(ILIKE.NE.1.AND.ILIKE.NE.4)GO TO 12
      PRINT*,"PLOT OF THE MINIMUM SING. VAL. OF CONT. MAT."
      CALL TITLES(ID1(13))
      PRINT*,"PLOT OF THE MAXIMUM SING. VAL. OF CONT. MAT."
      CALL TITLES(ID2(13))
12    IF(ILIKE.NE.2.AND.ILIKE.NE.4)GO TO 13
      PRINT*,"PLOT OF THE MINIMUM SING. VAL. OF OBS. MAT."
      CALL TITLES(ID3(13))
      PRINT*,"PLOT OF THE MAXIMUM SING. VAL. OF OBS. MAT."
      CALL TITLES(ID4(13))
13    IF(ILIKE.NE.1.AND.ILIKE.NE.4)GO TO 14
      PRINT*,"PLOT OF THE CONDITION NUMBER OF CONT. MAT."
      CALL TITLES(ID5(13))
14    IF(ILIKE.NE.2.AND.ILIKE.NE.4)GO TO 15
      PRINT ,"PLOT OF THE CONDITION NUMBER OF OBS. MAT."
      CALL TITLES(ID6(13))
15    IF(ILIKE.NE.3.AND.ILIKE.NE.4)GO TO 201
      PRINT*,"PLOT OF THE MAXIMUM SING. VAL. OF HANKEL MAT."
      CALL TITLES(ID7(13))
      PRINT*,"PLOT OF THE MINIMUM SING. VAL. OF HANKEL MAT."
      CALL TITLES(ID8(13))
      PRINT*,"PLOT OF THE CONDITION NUMBER OF HANKEL MAT."
```

AD-A080 371 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: --ETC(U)
DEC 79 J R MCCLENDON
UNCLASSIFIED AFIT/GE/EE/79-22 NL

3 OF 3
AD
A080371

END
DATE
FILMED
3-80
DDC

```
      CALL TITLES(ID9(13))
260   PRINT*," "
      PRINT*,"NOTE: THE CALCULATION DELTA FOR THIS ALGORITHM"
      PRINT*,"IS THE INITIAL TIME. THEREFORE THE INITIAL TIME MUST"
      PRINT*,"BE LESS THAN ONE. THE FINAL TIME MUST BE WITHIN"
      PRINT*,"30 STEPS OF THE INITIAL TIME"
      PRINT*,"PLEASE ENTER THE BEGINNING SAMPLING TIME"
      PRINT*,"AND THE FINAL SAMPLING TIME > "
      READ*,TINIT,TF
      TDEL=TINIT
      IF(SUP)PRINT*,"EVEN WITH LISTING SUPPRESSED THIS OPTION"
      IF(SUP)PRINT*,"TAKES TIME--DEPENDENT ON NUMBER OF POINTS"
      NT=(TF-TINIT)/TDEL
      T=TINIT
      TMAT(1)=T
      CALL DKREET(F,G,N,MB,NB,WKAREA,A,B,TMAT(1))
      CALL CONBTY(F,G,MB,NB,N,RMOD,WORKAREA,WINMED,N1)
      CALL OBS(MC,NC,N,F,WINMED,WORKAREA,RMOB,C,N2)
      DO 2III=1,N
      DO 2JJ=1,N1
      RMCC(JJ,III)=RMOD(III,JJ)
2     CONTINUE
      CALL LSVALR(RMCC,N1,N,30,30,1,WINMED,SIGC,UU,VV)
      CALL LSVALP(RMOB,N2,N,30,30,0,WINMED,SIGO,UU,VV)
      IF(ILIKE.NE.3.AND.ILIKE.NE.4)GO TO 2006
      CALL OVERLAY(4HMIMO,4,1)
2006  CALL VSORTA(SIGC,N)
      CALL VSORTA(SIGO,N)
      SCMAX(1)=SIGC(JWAS,1)
      SCMIN(1)=SIGC(IWAS,1)
      SOMAX(1)=SIGO(JWAS,1)
      SOMIN(1)=SIGO(IWAS,1)
      CONDC(1)=SCMIN(1)/SCMAX(1)
      CONDO(1)=SOMIN(1)/SOMAX(1)
      IF(SUP)GO TO 1000
      IF(ILIKE.EQ.2.OR.ILIKE.EQ.3)GO TO 3005
      PRINT*,"THE SINGULAR VALUES OF THE CONTROLLABILITY"
      PRINT*,"MATRIX FOR TIME > ",TMAT(1)," ARE "
      CALL PRINTM(N,1,SIGC)
      PRINT*," "
```

```
3000  IF(ILIKE.EQ.1.OR.ILIKE.EQ.3)GO TO 1000
      PRINT*,"THE SINGULAR VALUES OF THE OBSERVABILITY"
      PRINT*,"MATRIX FOR TIME > ",TMAT(1)," ARE "
      CALL PRINTM(N,1,SIGO)
1000  CONTINUE
      T=T+TDEL
      DO 200 I=1,N
      DO 200 J=1,N
      UC(I,J)=F(I,J)
      UCTRAN(I,J)=WKAREA(I,J)
      VOUT(I,J)=F(I,J)
      VOUT(I,I)=F(I,I)+1.
200   CONTINUE
      TMAT(2)=T
      CALL MATIA(UCTRAN,VOUT,N,N,N,VOUT)
      CALL MMULT(F,UC,WOKAREA,N,N,N)
      DO 7 LL=1,N
      DO 7 NN=1,N
      F(LL,NN)=WOKAREA(LL,NN)
7     CONTINUE
      CALL MAT1(VOUT,B,N,NB,NB,S)
      CALL CONBTY(F,G,MB,NB,N,RPDD,WOPAREA,WINMED,N1)
      CALL OBS(MC,NC,N,F,WINMED,WOKAREA,RMOB,C,N2)
      DO 3 II=1,N
      DO 3 JJ=1,N1
      RMOC(JJ,II)=RMOB(II,JJ)
3     CONTINUE
      CALL LSVALF(RMOC,N1,N,30,30,0,WKAREA,SIGC,UU,VV)
      CALL LSVALF(RMOB,N2,N,30,30,0,WKAREA,SIGO,UU,VV)
      IF(ILIKE.NE.3.AND.ILIKE.NE.4)GO TO 2007
      CALL OVERLAY(4HMIMO,4,2)
2007  CALL VSORTA(SIGC,N)
      CALL VSORTA(SIGO,N)
      SCMAX(2)=SIGC(JWAS,1)
      SOMAX(2)=SIGO(JWAS,1)
      SCMIN(2)=SIGC(IWAS,1)
      SOMIN(2)=SIGO(IWAS,1)
      CONDC(2)=SIGC(IWAS,1)/SIGC(JWAS,1)
      CONDO(2)=SIGO(IWAS,1)/SIGO(JWAS,1)
      IF(SUP)GO TO 2000
      IF(ILIKE.EQ.2.OR.ILIKE.EQ.3)GO TO 3007                          009300
      PRINT*,"THE SINGULAR VALUES OF THE CONTROLLABILITY"             009310
      PRINT*,"MATRIX FOR TIME> ",TMAT(2)," ARE "                      009320
      CALL PRINTM(N,1,SIGC)                                           009330
3007  IF(ILIKE.EQ.1.OR.ILIKE.EQ.3)GO TO 2006                          009340
      PRINT*,"THE SINGULAR VALUES OF THE OBSERVABILITY"               009350
      PRINT*,"MATRIX FOR TIME>", TMAT(2),"ARE "                       009360
      CALL PRINTM(N,1,SIGO)                                           009370
2006  CONTINUE                                                        009380
      T=T+TDEL                                                        009390
      DO 265 I=3,NT                                                   009400
      TMAT(I)=T                                                       009410
      CALL MAT1(F,UCTRAN,N,N,N,WINMED)                                009420
      CALL MMULT(F,UC,WORAREA,N,N,N)                                  009430
      DO 6 LL=1,N                                                     009440
      DO 6 NN=1,N                                                     009450
      F(LL,NN)=WORAREA(LL,NN)                                         009460
6     CONTINUE                                                        009470
      DO 265 L=1,N                                                    009480
      DO 265 J=1,N                                                    009490
      VOUT(L,J)=VOUT(L,J)+WINMED(L,J)                                 009500
265   CONTINUE                                                        009510
      CALL MAT1(VOUT,B,N,MB,NB,S)                                     009520
      CALL COMBTY(F,G,MB,NB,N,RMOD,WORAREA,WINMED,N1)                 009530
      CALL OBS(MC,NC,N,F,WINMED,WORAREA,RMOP,C,N2)                    009540
      DO 4 II=1,N                                                     009550
      DO 4 JJ=1,N1                                                    009560
      RMOC(JJ,II)=RMOD(II,JJ)                                         009570
4     CONTINUE                                                        009580
      CALL LSVALR(RMOC,N1,N,30,30,0,WKAREA,SIGC,UU,VV)                009590
      CALL LSVALR(RMOB,N2,N,30,30,0,WKAREA,SIGO,UU,VV)                009600
      IF(ILIKE.NE.3.AND.ILIKE.NE.4)GO TO 2008                         009610
      CALL OVERLAY(4HMIMO,4,3)                                        009620
2008  CALL VSORTA(SIGC,N)                                             009630
      CALL VSORTA(SIGO,N)                                             009640
      SCMAX(I)=SIGC(JWAS,1)                                           009650
      SOMAX(I)=SIGO(JWAS,1)                                           009660
      SCMIN(I)=SIGC(IWAS,1)                                           009670
      SOMIN(I)=SIGO(IWAS,1)                                           009680
      CONDC(I)=SIGC(IWAS,1)/SIGC(JWAS,1)                              009690
      CONDO(I)=SIGO(IMAS,1)/SIGO(JMAS,1)
      IF(SUP)GO TO 3003
      IF(ILIKE.EQ.2.OR.ILIKE.EQ.3)GO TO 3003
      PRINT*,"THE SINGULAR VALUES OF THE CONTROLLABILITY"
      PRINT*,"MATRIX FOR TIME> ",TMAT(I)," ARE "
      CALL PRINTM(N,1,SIGC)
      PRINT*," "
3003  IF(ILIKE.EQ.1.OR.ILIKE.EQ.3)GO TO 3000
      PRINT*,"THE SINGULAR VALUES OF THE OBSERVABILITY"
      PRINT*,"MATRIX FOR TIME> ",TMAT(I)," ARE "
      CALL PRINTM(N,1,SIGO)
3000  CONTINUE
      T=T+TDEL
206   CONTINUE
      CALL PLOTS(IDATA,NUM,MJM)
      CALL PLOT(0.-4.,-3)
      CALL PLOT(0.,0.63,-3)
      IF(ILIKE.NE.1.AND.ILIKE.NE.4)GO TO 2001
      CALL HGRAPH(TMAT,SCMIN,NT,ID1,1,0,0)
      CALL HGRAPH(TMAT,SCMAX,NT,ID2,1,0,0)
2001  IF(ILIKE.NE.2.AND.ILIKE.NE.4)GO TO 2002
      CALL HGRAPH(TMAT,SOMIN,NT,ID3,1,0,0)
      CALL HGRAPH(TMAT,SOMAX,NT,ID4,1,0,0)
2002  IF(ILIKE.NE.1.AND.ILIKE.NE.4)GO TO 2003
      CALL HGRAPH(TMAT,CONDC,NT,ID5,1,0,0)
2003  IF(ILIKE.NE.2.AND.ILIKE.NE.4)GO TO 2004
      CALL HGRAPH(TMAT,CONDO,NT,ID6,1,0,0)
2004  IF(ILIKE.NE.3.AND.ILIKE.NE.4)GO TO 2005
      CALL HGRAPH(TMAT,HMAX,NT,ID7,1,0,0)
      CALL HGRAPH(TMAT,HMIN,NT,ID8,1,0,0)
      CALL HGRAPH(TMAT,HCOND,NT,ID9,1,0,0)
2005  CALL PLOTE(MMUUM)
      END
C*************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C*************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 2 I=1,M,1
      WRITE (5,10) I,(DMAT(I,J),J=1,N,1)
```

```
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C*************************************************************
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.0D0
      DO 1 J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3 J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C*************************************************************
      SUBROUTINE PRINTM1(M,N,EMAT)
      DIMENSION EMAT(30,10)
      DO 20I=1,M,1
      WRITE (6,10) I,(EMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE FINDS EXP(AMAT*T) AND ITS INTEGRAL
C     VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C*************************************************************
      SUBROUTINE DISKRET(N,A,DEL,EA,EAINT,NT)
      DIMENSION EA(10,10),EAINT(10,10),A(10,10),Z(10,10),
     CHINNER(10,10),CBAR(10,10)
      DO 10 I=1,N
      DO 10 J=1,N
      EAINT(I,J)=0.000
      EAINT(I,I)=DEL
      Z(I,J)=EAINT(I,J)
130   CONTINUE
      DO 410 K=2,NT
      DO 150 I=1,N
      DO 150 J=1,N
      CBAR(I,J)=A(I,J)*(DEL/FLOAT(K))
150   CONTINUE
      CALL MMULT(Z,CBAR,WINMED,N,N,N)
      DO 160 I=1,N
      DO 160 J=1,N
      Z(I,J)=WINMED(I,J)
      EAINT(I,J)=EAINT(I,J)+WINMED(I,J)
160   CONTINUE
410   CONTINUE
      CALL MMULT(A,EAINT,Z,N,N,N)
      DO 500 I=1,N
      DO 500 J=1,N
      EA(I,J)=0.0D0
      EA(I,I)=1.0D0
500   CONTINUE
      DO 600 I=1,N
      DO 600 J=1,N
      EA(I,J)=EA(I,J)+Z(I,J)
600   CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE ACCEPTS EXP(AMAT*T) AND ITS INTEGRAL
C     AND FORMS THE DISCRETE F MATRIX AND G MATRIX
C*****************************************************************
      SUBROUTINE DKREET(F,G,N,NB,NB,WKAREA,A,B,T)
      DIMENSION F(10,10),G(10,11),WKAREA(10,10),A(10,10)
      DIMENSION B(10,10)
      CALL DISKRETE(N,A,T,F,WKAREA,50)
      CALL MMULT(WKAREA,B,G,N,N,NB)
      RETURN
      END
C*****************************************************************
```

```
C     THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C     MATRIX
C**************************************************************
      SUBROUTINE CONBTY(F,G,MB,NB,N,RMOD,WORAREA,WINMED,N1)
      DIMENSION F(10,10),G(10,10),RMOD(10,30),WINMED(10,10)
      DIMENSION WORAREA(10,10)
      DO 100 I=1,MB
      DO 100 J=1,NB
      RMOD(I,J)=G(I,J)
100   CONTINUE
      N1=NB
      IF(N.EQ.1)GO TO 150
      CALL MMULT(F,G,WINMED,N,N,NB)
      DO 120 I=1,MB
      DO 120 J=1,NB
      K=J
      N1=NB
      RMOD(I,NB+J)=WINMED(I,J)
120   CONTINUE
      N1=N1+K
      IF(N.EQ.2)GO TO 150
      NN=N-1
      DO 140 L=2,NN,1
      CALL MMULT(F,WINMED,WORAREA,N,N,NB)
      DO 130 I=1,MB,1
      DO 130 J=1,NB,1
      RMOD(I,N1+J)=WORAREA(I,J)
      WINMED(I,J)=WORAREA(I,J)
      K=J
130   CONTINUE
      N1=N1+K
140   CONTINUE
150   RETURN
      END
C**************************************************************
C     THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX
C**************************************************************
      SUBROUTINE OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)
      DIMENSION WINMED(10,10),WORAREA(10,10),C(10,10)
      DIMENSION F(10,10),RMOB(30,10)
      DO 200 I=1,MC
      DO 200 J=1,NC
      RMOB(I,J)=C(I,J)
      N2=MC
200   CONTINUE
      IF(N.EQ.1)GO TO 250
      CALL MMULT(C,F,WINMED,MC,NC,N)
      DO 220 I=1,MC
      DO 220 J=1,NC
      K=I
      N2=MC
      RMOB(N2+I,J)=WINMED(I,J)
220   CONTINUE
      N2=N2+K
      IF(N.EQ.2)GO TO 250
      NN=N-1
      DO 240 L=2,NN,1
      CALL MMULT(WINMED,F,WORKAREA,MC,NC,N)
      DO 230 I=1,MC
      DO 230 J=1,NC
      RMOB(N2+I,J)=WORKAREA(I,J)
      WINMED(I,J)=WORKAREA(I,J)
      K=I
230   CONTINUE
      N2=N2+K
240   CONTINUE
250   RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE PLOTS A HORIZONTAL GRAPH
C*****************************************************************
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)
      DIMENSION X(1),Y(1),ID(1)   $IF(NO.EQ.2)GO TO 30
      IF(NO.LT.0)GO TO 10
      CALL SCALE(Y,7.,N,1)  $CALL SCALE(Y,5.,N,1)
10    CALL PLOT(8.5,0.,-3) $ CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2) $ CALL PLOT(-7.15,9.65,2)
      IF(ID(1).EQ.000)GO TO 25
      CALL PLOT(-7.05,9.55,3) $ CALL PLOT(-7.05,7.55,2)
      DO 20I=1,7,2                                                      011700
20    CALL SYMBOL(I*.1-6.9,7.35,.07,ID(I),90.,20)                       011710
      CALL PLOT(-7.05,7.55,3) $CALL PLOT(-6.05,7.55,2)                  011720
      CALL PLOT(-6.05,9.55,2) $ CALL PLOT(-7.05,9.55,2)                 011730
      CALL PLOT(-7.15,9.65,3)                                           011740
25    CALL PLOT(-1.35,9.65,2) $ CALL PLOT(-1.35,1.35,2)                 011750
      CALL SYMBOL(-6.55,1.15,.1,ID(13),0.,50)                           011760
      CALL AXIS(-1.65,2.1,ID(9),-20,7.,90.,X(N+1),X(N+2))               011770
      CALL AXIS(-1.65,2.1,ID(11),20,5.,180.,Y(N+1),Y(N+2))              011780
30    Y(N+2)=-Y(N+2)                                                    011790
      X(N+1)=X(N+1)-2.1*X(N+2) $ Y(N+1)=Y(N+1)+1.65*Y(N+2)              011800
      CALL LINE(Y,X,N,1,NP,NS)                                          011810
      X(N+1)=X(N+1)+2.1*X(N+2) $ Y(N+1)=Y(N+1)-1.65*Y(N+2)              011820
      Y(N+2)=-Y(N+2)                                                    011830
      RETURN   $   END                                                  011840
C*********************************************************************   011850
C     THIS SUBROUTINE PLOTS A VERTICAL GRAPH                            011860
C*********************************************************************   011870
      SUBROUTINE VGRAPH(X,Y,N,ID,NO,NP,NS)                              011880
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2)GO TO 30                   011890
      IF(NO.LT.0)GO TO 10                                               011900
      CALL SCALE(X,4.5,N,1) $ CALL SCALE(Y,7.,N,1)                      011910
10    CALL PLOT(8.5,0.,-3) $ CALL PLOT(0.,11.,3)                        011920
      CALL PLOT(-1.35,1.35,3)                                           011930
      CALL PLOT(-7.15,1.35,2) $ CALL PLOT(-7.15,9.65,2)                 011940
      CALL PLOT(-1.35,9.65,2) $ IF(ID(1).EQ.6HU)GO TO 25                011950
      CALL PLOT(-1.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)                 011960
      DO 20I=1,7,2                                                      011970
20    CALL SYMBOL(-3.15,9.4-I*.10,.07,ID(I),0.,10)                      011980
      CALL PLOT(-3.45,9.55,3) $ CALL PLOT(-3.45,8.95,2)                 011990
      CALL PLOT(-1.45,8.95,2) $ CALL PLOT(-1.45,9.55,2)                 012000
      CALL PLOT(-1.35,9.65,3)                                           012010
25    CALL PLOT(-1.35,1.35,2)                                           012020
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,30)                           012030
      CALL AXIS(-6.4,1.65,ID(9),-20,4.5,0.,X(N+1),X(N+2))               012040
      CALL AXIS(-6.4,1.65,ID(11),20,7.0,90.,Y(N+1),Y(N+2))              012050
30    X(N+1)=X(N+1)+6.4*X(N+2) $ Y(N+1)=Y(N+1)-1.65*Y(N+2)              012060
      CALL LINE(X,Y,N,1,NP,NS)                                          012070
      X(N+1)=X(N+1)-6.4*X(N+2) $ Y(N+1)=Y(N+1)+1.65*Y(N+2)              012080
      RETURN $     END                                                  012090
```

```
C*************************************************************
C     THIS SUBROUTINE ALLOWS VARIABLE TITLES
C*************************************************************
      SUBROUTINE TITLES(TITLE)
      INTEGER TITLE
      DIMENSION INK(80),TITLE(5)
1111  DO 10 J=1,5
10    TITLE(J)=0
      PRINT*,"> (--------ENTER TITLE  (50 CHARACTERS MAX)--------) <"
      PRINT*,"> "
      READ 1000,INK
1000  FORMAT(80R1)
      NSP=II=0
1     II=II+1
      IF(II.GT.80)GO TO 60
      NSP=NSP+1
      IF(INK(II).NE.1R ) NSP=0
      IF(II.GT.50)GO TO 1
      JJ=(II-1)/10+1
      TITLE(JJ)=TITLE(JJ)*64+INK(II)
      GO TO 1
60    IF(NSP.GE.30)GO TO 65
      PRINT*,"TITLE TRUNCATED TO 50 CHARACTERS"
      PRINT 1001,TITLE
      GO TO 1111
65    LETTERS=80-NSP
      PRINT 1001,TITLE
1001  FORMAT(3X,5A10)
      RETURN
      END
C*************************************************************
      OVERLAY(4,1)
C*************************************************************
      PROGRAM MANK1
      LOGICAL SUP
      COMMON/MEMO2/THAT(100),SCMAX(100),SCMIN(100),SOMAX(100),SOMIN(100)
     +,CONDS(100),CONDO(100),HMAX(100),HMIN(100),HCOND(100)
      COMMON/MEMO3/RMOB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,UU(30,30),
     +VV(30,30),HHKL(30,30),HHKLT(30,30)
      COMMON/MEMO4/HORAREA(10,10),HRAREA(10,10),WINMED(10,10)
      COMMON/MIMOF/A(10,10),M,N,B(10,10),MB,NB,F(10,10),
     +G(10,10),C(10,10),MC,NC
      COMMON/PLOTF1/1,SIVS(30,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      CALL VMULFF(HMOD,HMOD,N2,N,N1,30,10,HNKL,30,IER)
      IF(N2.GE.N1)CALL LSVALR(HNKL,N2,N1,30,30,0,VV,SIVS,UU,VV)
      DO 5II=1,N2
      DO 5JJ=1,N1
      HNKLT(JJ,II)=HNKL(II,JJ)
5     CONTINUE
      IF(N2.LT.N1)CALL LSVALR(HNKLT,N1,N2,30,30,0,VV,SIVS,UU,VV)
      IF(N2.GE.N1)CALL VSORTA(SIVS,N1)
      IF(N2.LT.N1)CALL VSORTA(SIVS,N2)
      IF(N2.GE.N1)NUM=N1
      IF(N2.LT.N1)NUM=N2
      IWAS1=IWAS
      JWAS1=JWAS
      IF(SUP)GO TO 4000
      PRINT ,"THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME ",
     +TMAT(1)," ARE > "
      IF(N2.GE.N1)CALL PRINTM2(N1,1,SIVS)
      IF(N2.LT.N1)CALL PRINTM2(N2,1,SIVS)
4000  CONTINUE
      IF(N2.GE.N1)HMAX(1)=SIVS(JWAS1,1)
      IF(N2.LT.N1)HMAX(1)=SIVS(JWAS1,1)
      HMIN(1)=SIVS(IWAS1,1)
      HCOND(1)=HMIN(1)/HMAX(1)
      END
C************************************************************
C     THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SINGULAR VALUES
C************************************************************
      SUBROUTINE PRINTM2(M,N,DMAT)
      DIMENSION DMAT(30,1)
      DO 20I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C################################################################
      OVERLAY(4,2)
```

```
C*******************************************************************
      PROGRAM HANK2
      LOGICAL SUP
      COMMON/MIMO2/TMAT(100),SOMAX(100),SOMIN(100),SOMAX(100),SOMIN(100)
     +,CONDC(100),CONDO(100),HMAX(100),HMIN(100),HCOND(100)
      COMMON/MIMO2/RMOB(30,10),RMOD(10,30),RMOC(30,10),N1,N2,UU(30,30),
     +VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO /WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO /A(10,10),M,N,B(10,10),MB,NB,F(10,10),
     +G(10,10),C(10,10),MC,NC
      COMMON/PLOTS1/I,SIVS(30,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      CALL VMULFF(RMOB,RMOD,N2,N,N1,30,10,HNKL,30,IER)
      IF(N2.GE.N1)CALL LSVALF(HNKL,N2,N1,30,30,0,VV,SIVS,UU,VV)
      DO 5II=1,N2
      DO 5JJ=1,N1
      HNKLT(JJ,II)=HNKL(II,JJ)
5     CONTINUE
      IF(N2.LT.N1)CALL LSVALF(HNKLT,N1,N2,30,30,0,VV,SIVS,UU,VV)
      IF(N2.GE.N1)CALL VSORTA(SIVS,N1)
      IF(N2.LT.N1)CALL VSORTA(SIVS,N2)
      IF(SUP)GO TO 5000
      PRINT ,"THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME ",
     +TMAT(2)," ARE > "
      IF(N2.GE.N1)CALL PRINTM2(N1,1,SIVS)
      IF(N2.LT.N1)CALL PRINTM2(N2,1,SIVS)
5000  CONTINUE
      IF(N2.GE.N1)HMAX(2)=SIVS(JWAS1,1)
      IF(N2.LT.N1)HMAX(2)=SIVS(JWAS1,1)
      HMIN(2)=SIVS(IWAS1,1)
      HCOND(2)=HMIN(2)/HMAX(2)
      END
C*******************************************************************
C      THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SINGULAR VALUES
C*******************************************************************
      SUBROUTINE PRINTM2(M,N,DMAT)
      DIMENSION DMAT(30,1)
      DO 20I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
```

```
      RETURN
      END
C############################################################
      OVERLAY(4,3)
C############################################################
      PROGRAM HANK3
      LOGICAL SUP
      COMMON/HIMO2/TMAT(100),SCMAX(100),SCMIN(100),SOMAX(100),SOMIN(100)
     +,CONDC(100),CONDO(100),HMAX(100),HMIN(100),HCOND(100)
      COMMON/HIMO2/RMOB(30,10),RMOD(10,30),RMOC(30,10),N1,N2,UU(30,30),
     +VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/HIMO2/WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/HIMO2/A(10,10),M,N,B(10,10),MB,NB,F(10,10),
     +G(10,10),C(10,10),MC,NC
      COMMON/PLOTS1/1,SIVS(30,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      CALL VMULFF(RMOB,RMOD,N2,N,N1,30,10,HNKL,30,IER)
      IF(N2.GE.N1)CALL LSVA_R(HNKL,N2,N1,30,30,0,VV,SIVS,UU,VV)
      DO 5 II=1,N2
      DO 5 JJ=1,N1
      HNKLT(JJ,II)=HNKL(II,JJ)
5     CONTINUE
      IF(N2.LT.N1)CALL LSVA_R(HNKLT,N1,N2,30,30,0,VV,SIVS,UU,VV)
      IF(N2.GE.N1)CALL VSORTA(SIVS,N1)
      IF(N2.LT.N1)CALL VSORTA(SIVS,N2)
      IF(SUP)GO TO 6000
      PRINT ,"THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME ",
     +TMAT(I)," ARE > "
      IF(N2.GE.N1)CALL PRINTM2(N1,1,SIVS)
      IF(N2.LT.N1)CALL PRINTM2(N2,1,SIVS)
6000  CONTINUE
      IF(N2.GE.N1)HMAX(I)=SIVS(JWAS1,1)
      IF(N2.LT.N1)HMAX(I)=SIVS(JWAS1,1)
      HMIN(I)=SIVS(IWAS1,1)
      HCOND(I)=HMIN(I)/HMAX(I)
      END
C***********************************************************
C     THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SINGULAR VALUES
C***********************************************************
      SUBROUTINE PRINTM2(M,N,DMAT)
      DIMENSION DMAT(30,1)
      DO 20I=1,M,1                                                      013700
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)                                 013710
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))                                 013720
20    CONTINUE                                                          013730
      RETURN                                                            013740
      END                                                               013750
C*******************************************************************   013760
      OVERLAY(7,0)                                                      013770
C*******************************************************************   013780
C     THIS PROGRAM FINDS THE CONTINUOUS TIME CONTROLLABILITY            013790
C     AND OBSERVABILITY MATRICES.                                       013800
C                                                                       013810
C                                                                       013820
      PROGRAM CONTIME                                                   013830
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 013840
      COMMON/MIMO7/RMOB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,UU(30,30)  013850
     *,VV(30,30),HNKL(30,30),HJKLT(30,30)                               013860
      COMMON/MIMO /WORAREA(10,10),WKAREA(10,10),WINMED(10,10)           013870
      COMMON/MIMO /A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,1001 3880
     *),MC,NC                                                           013890
      CALL CONBTY(A,B,MB,NB,N,RMOD,WORAREA,WINMED,N1)                   013900
      CALL OBS(MC,NC,N,A,WINMED,WORAREA,RMOB,C,N2)                      013910
      PRINT*,"THE CONTINUOUS TIME CONTROLLABILITY MATRIX IS"            013920
      CALL PRINTM(N,N1,RMOD)                                            013930
      PRINT*,"THE CONTINUOUS TIME OBSERVABILITY MATRIX IS"              013940
      CALL PRINTM1(N2,N,RMOB)                                           013950
      END                                                               013960
C*******************************************************************   013970
C     THIS SUBROUTINE PRINTS OUT MATRICES                               013980
C*******************************************************************   013990
      SUBROUTINE PRINTM(M,N,DMAT)                                       014000
      DIMENSION DMAT(10,10)                                             014010
      DO 20I=1,M,1                                                      014020
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)                                 014030
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))                                 014040
20    CONTINUE                                                          014050
      RETURN                                                            014060
      END                                                               014070
C*******************************************************************   014080
C     THIS SUBROUTINE PRINTS OUT THE OBSERVABILITY MATRIX               014090
```

```
C*********************************************************************
      SUBROUTINE PRINTM1(M,N,EMAT)
      DIMENSION EMAT(30,10)
      DO 20I=1,M,1
      WRITE(6,10) I,(EMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE15.4,1X)))
20    CONTINUE
      RETURN
      END
C*********************************************************************
C     THIS SUBROUTINE FINDS THE CONTINUOUS TIME CONTROLLABILITY MATRIX
C*********************************************************************
      SUBROUTINE CONBTY(F,G,NB,NB,N,RMOD,WORAREA,WINMED,N1)
      DIMENSION F(10,10),G(10,10),RMOD(10,30),WINMED(10,10)
      DIMENSION WORAREA(10,10)
      DO 110 I=1,NB
      DO 110 J=1,NB
      RMOD(I,J)=G(I,J)
110   CONTINUE
      N1=NB
      IF(N.EQ.1)GO TO 150
      CALL MMULT(F,G,WINMED,N,N,NB)
      DO 120 I=1,NB
      DO 120 J=1,NB
      K=J
      N1=NB
      RMOD(I,N1+J)=WINMED(I,J)
120   CONTINUE
      N1=N1+K
      IF(N.EQ.2)GO TO 150
      NN=N-1
      DO 140 L=2,NN,1
      CALL MMULT(F,WINMED,WORAREA,N,N,NB)
      DO 130 I=1,NB,1
      DO 130 J=1,NB,1
      RMOD(I,N1+J)=WORAREA(I,J)
      WINMED(I,J)=WORAREA(I,J)
      K=J
130   CONTINUE
      N1=N1+K
```

```
140   CONTINUE                                                                  014500
150   RETURN                                                                    014510
      END                                                                       014520
C*************************************************************                 014530
C     THIS SUBROUTINE FINDS THE CONTINUOUS OBSEPVABILITY MATRIX                014540
C*************************************************************                 014550
      SUBROUTINE OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)                        014560
      DIMENSION WINMED(10,10),WORAREA(10,10),C(10,10)                           014570
      DIMENSION F(10,10),RMOB(30,10)                                            014580
      DO 210 I=1,MC                                                             014590
      DO 210 J=1,NC                                                             014600
      RMOB(I,J)=C(I,J)                                                          014610
      N2=MC                                                                     014620
210   CONTINUE                                                                  014630
      IF(N.EQ.1)GO TO 250                                                       014640
      CALL MMULT(F,F,WINMED,MC,NC,N)                                            014650
      DO 220 I=1,MC                                                             014660
      DO 220 J=1,NC                                                             014670
      K=I                                                                       014680
      N2=MC                                                                     014690
      RMOB(N2+I,J)=WINMED(I,J)                                                  014700
220   CONTINUE                                                                  014710
      N2=N2+K                                                                   014720
      IF(N.EQ.2)GO TO 250                                                       014730
      NN=N-1                                                                    014740
      DO 240 L=2,NN,1                                                           014750
      CALL MMULT(WINMED,F,WORAREA,MC,NC,N)                                      014760
      DO 230 I=1,MC                                                             014770
      DO 230 J=1,NC                                                             014780
      RMOB(N2+I,J)=WORAREA(I,J)                                                 014790
      WINMED(I,J)=WORAREA(I,J)                                                  014800
      K=I                                                                       014810
230   CONTINUE                                                                  014820
      N2=N2+K                                                                   014830
240   CONTINUE                                                                  014840
250   RETURN                                                                    014850
      END                                                                       014860
C*************************************************************                 014870
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES                                   014880
C*************************************************************                 014890
```

```
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.0
      DO 1 J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
C*****************************************************************
      OVERLAY(5,0)
C*****************************************************************
C     THIS PROGRAM USES A MULTITUDE OF SUBROUTINES FOUND IN
C     LINPACK(SIAM PUBLISHING 1979) TO ESTIMATE THE CONDITION
C     NUMBER WITH RESPECT TO INVERSION OF A SQUARE MATRIX.
C     WHERE THE CONDITION NUMBER IS DEFINED AS IIAII IIAINVERSEII
C     WHERE II II DENOTES THE IS THE NORM OF A MATRIX.
C
C
      PROGRAM RCOEST
      DIMENSION IPVT(30),ZZ(30),PMAT(10,10)
      COMMON/MIMO2/AMOB(30,10),AMOD(10,30),AMCC(30,10),N1,N2,UU(30,30)
     +,VV(30,30),PNKL(30,30),HNKLT(30,30)
      COMMON/MIMO /A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10
     +),MC,NC
      COMMON/MIMO7/BBAR(10,10),SIGO2IN(10,10),Z(10,10),CBAR(10,10),
     +TINV(10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      COMMON/MIMOF/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     +VHTRAN(10,10),SIGO2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      RCOND=0.0E0
5     PRINT ,"THIS OPTION ESTIMATES THE CONDITION NUMBER OF A"
      PRINT ,"SQUARE MATRIX.  ENTER THE NUMBER CORRESPONDING"
      PRINT ,"TO THE DESIRED INPUT MATRIX"
      PRINT ,"1---------AMAT(ORIGINAL)"
      PRINT ,"2---------AMAT(BALANCED)"
      PRINT ,"3---------F"
      PRINT*,"4---------ENTER YOUR OWN"
      PRINT*,"5---------MATRIX TO HELP.EVALUATE ORDER REDUCTION"
      PRINT*,"6---------HANKEL MAT.(MUST BE CREATED BY OPT. 3)"
      PRINT*,"ENTER > "
      READ*,NEX
      DO 1IO=1,10
      DO 1JO=1,10
      CREAL(IO,JO)=AREAL(IO,JO)
      BREAL(IO,JO)=A(IO,JO)
1     CONTINUE
      GO TO (10,20,30,40,60,70),NEX
10    CALL SGECO(BREAL,10,M,IPVT,RCOND,ZZ)
      GO TO 50
20    CALL SGECO(CREAL,10,M,IPVT,RCOND,ZZ)
      GO TO 50
30    CALL SGECO(F,10,M,IPVT,RCOND,ZZ)
      GO TO 50
60    CALL SGECO(HINF,10,M,IPVT,RCOND,ZZ)
      GO TO 50
70    IF(N1.NE.N2)GO TO 42
      CALL SGECO(HNKL,30,N1,IPVT,RCOND,ZZ)
      GO TO 50
42    DO 45IB=1,N2
      DO 45JB=1,N1
      HNKLT(JB,IB)=HNKL(IB,JB)
45    CONTINUE
      IF(N1.LT.N2)CALL VMULFF(HNKLT,HNKL,N1,N2,N1,30,30,UU,30,IER)
      IF(N1.GT.N2)CALL VMULFF(HNKL,HNKLT,N2,N1,N2,30,30,UU,30,IER)
      IF(N1.LT.N2)CALL SGECO(UU,30,N1,IPVT,RCOND,ZZ)
      IF(N1.GT.N2)CALL SGECO(UU,30,N2,IPVT,RCOND,ZZ)
      GO TO 50
40    PRINT*,"ENTER THE ORDER OF THE SQUARE MATRIX > "
      READ*,MA
      CALL READM(MA,MA,PMAT)
      CALL PRINTM(MA,MA,PMAT)
      CALL MISTAKE(MA,MA,PMAT)
      CALL SGECO(PMAT,10,MA,IPVT,RCOND,ZZ)
50    RCOND=1./RCOND
      PRINT*," "
      PRINT*,"THE ESTIMATED CONDITION NUMBER IS RCOND= ",RCOND
      END
C*****************************************************
C     THIS SUBROUTINE READS IN A GIVEN MATRIX
C*****************************************************
      SUBROUTINE READM(M,N,RMAT)
      DIMENSION RMAT(10,10)
      PRINT*,"PLEASE ENTER THE MATRIX BY ROWS"
100   DO 105 I=1,M,1
10    WRITE(6,15) I
15    FORMAT(1H0,"ROW(",I2,")=")
      READ*,(RMAT(I,J),J=1,N,1)
105   CONTINUE
110   RETURN
      END
C*****************************************************
C     THIS SUBROUTINE ALLOWS USER TO CHANGE ERRONEOUS DATA
C*****************************************************
      SUBROUTINE MISTAKE(M,N,RMAT)
      DIMENSION RMAT(10,10)
      MIST=2
      PRINT*,"DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) > "
      READ*,INDIC
      IF(INDIC.EQ.2)GO TO 2
3     PRINT*,"TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
      MIST=1
      PRINT*,"THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
      PRINT*,"MATRIX ELEMENT SEPARATED BY COMMAS > "
      READ*,MM,NN,CORREC
      RMAT(MM,NN)=CORREC
      PRINT*,"ANY MORE(1=YES,2=NO) > "
      READ*,INDU
      GO TO 3
2     IF(MIST.EQ.1)CALL PRINTM(M,N,RMAT)
      RETURN
      END
C*****************************************************
C     THIS SUBROUTINE PRINTS OUT A GIVEN MATRIX
C*****************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20 I=1,M,1
      WRITE (6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C***************************************************************
      OVERLAY(7,0)
C***************************************************************
C     THIS PROGRAM FINDS THE SINGULAR VALUES OF A USER-SPECIFIED
C     MATRIX.
C
C
      PROGRAM SIGV
      DIMENSION PFMAT(10,10),PPMATRN(10,10)
      COMMON/MIMO2/PMCB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,UU(30,30)
     *,VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO/WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10
     *),MC,NC
      COMMON/MIMO/BBAR(10,10),SIGUZIN(10,10),Z(10,10),CBAR(10,10),
     *TINV(10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      COMMON/MIMO/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     *VHTRAN(10,10),SIGC2(10,10),SISO2(10,10),VOUTRAN(10,10)
      COMMON/PLOTF1/I,SIVS(30,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      PRINT*,"THIS OPTION OBTAINS THE SINGULAR VALUES OF A GIVEN"
      PRINT*,"MATRIX--NEED NOT BE SQUARE.  "
      MPP=NPP=1
      PRINT*,"ENTER DESIRED CORRESPONDING NUMBER"
      PRINT*,"1---------AMAT(ORIGINAL)"
      PRINT*,"2---------AMAT(BALANCED)"
      PRINT ,"3---------F"
      PRINT*,"4---------ENTER YOUR OWN"
      PRINT*,"5---------DISCRETE HANKEL MATRIX"
      PRINT*,"ENTER > "
      READ*,NEX1
      PRINT*,"DO YOU WISH TO PRINT OUT THE "
      PRINT*,"SINGULAR VECTORS AS WELL?"
      PRINT*,"1=NO, 2=LEFT S.V., 3=RIGHT S.V., 4=BOTH"
      PRINT*,"ENTER > "
      READ*,IMAY
      GO TO (10,20,30,40,60),NEX1
10    CALL LSVALR(A,N,N,10,10,1,WKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,VHTRAN,N,N,10)
      GO TO 50
20    CALL LSVALR(AREAL,N,N,10,10,1,WKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,VHTRAN,N,N,10)
      GO TO 50
30    CALL LSVALR(F,N,N,10,10,1,WKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,VHTRAN,N,N,10)
      GO TO 50
60    DO 100 I=1,N2
      DO 100 J=1,N1
      HNKLT(J,I)=HNKL(I,J)
100   CONTINUE
      IF(N2.LE.N1)CALL LSVALR(HNKLT,N1,N2,30,30,1,HNKL,SIVS,UU,VV)
      IF(N2.GT.N1)CALL LSVALR(HNKL,N2,N1,30,30,1,HNKLT,SIVS,UU,VV)
      GO TO 50
40    PRINT*,"ENTER ROW DIMENSION OF INPUT MATRIX  > "
      READ*,MPP
      PRINT*,"ENTER COLUMN DIMENSION OF INPUT MATRIX > "
      READ*,NPP
      CALL READM(MPP,NPP,PPMAT)
      CALL PRINTM(MPP,NPP,PPMAT)
      CALL MISTAKE(MPP,NPP,PPMAT)
      IF(NPP.LT.MPP)GO TO 45
      DO 42 I=1,MPP
      DO 42 J=1,NPP
      PPMATRN(J,I)=PPMAT(I,J)
42    CONTINUE
      CALL LSVALR(PPMATRN,NPP,MPP,10,10,1,WKAREA,WINMED,UH,VHTRAN)
      GO TO 50
45    CALL LSVALR(PPMAT,MPP,NPP,10,10,1,WKAREA,WINMED,UH,VHTRAN)
50    PRINT*,"THE NON-ZERO SINGULAR VALUES ARE:   "
      IF(NEX1.EQ.5.AND.N2.LE.N1)CALL PRINTM1(N2,1,SIVS)
      IF(NEX1.EQ.5.AND.N2.GT.N1)CALL PRINTM1(N1,1,SIVS)
      IF(MPP.GT.0.AND.MPP.LE.NPP)CALL PRINTM(MPP,1,WINMED)
      IF(MPP.GT.0.AND.MPP.GT.NPP)CALL PRINTM(NPP,1,WINMED)
      IF(MPP.EQ.0.AND.NEX1.NE.5)CALL PRINTM(N,1,WINMED)
      IF(NEX1.EQ.5.AND.N2.LE.N1)CALL PRINTS1(IMAY,UU,VV,N1,N2,30)
      IF(NEX1.EQ.5.AND.N2.GT.N1)CALL PRINTS1(IMAY,UU,VV,N2,N1,30)
      IF(NEX1.EQ.6.AND.MPP.LE.NPP)CALL PRINTS(IMAY,UH,VHTRAN
     +,MPP,NPP,10)
      IF(NEX1.EQ.6.AND.MPP.GT.NPP)CALL PRINTS(IMAY,UH,VHTRAN
     +,MPP,NPP,10)
      END
      SUBROUTINE PRINTS1(IMAY,UU,VV,N1,N2,IRNK)
      DIMENSION UU(IRNK,1),VV(IRNK,1)
      IF(IMAY.EQ.2.OR.IMAY.EQ.4)PRINT*,"THE LEFT SINGULAR VECTORS"
      IF(IMAY.EQ.2.OR.IMAY.EQ.4)CALL PRINTM1(N1,N2,UU)
      IF(IMAY.EQ.3.OR.IMAY.EQ.4)PRINT*,"THE RIGHT SINGULAR VECTORS"
      IF(IMAY.EQ.3.OR.IMAY.EQ.4)CALL PRINTM1(N2,N2,VV)
      RETURN
      END
C*******************************************************
C     THIS SUBROUTINE PRINTS OUT SINGULAR VECTORS
C*******************************************************
      SUBROUTINE PRINTS(IMAY,UU,VV,N1,N2,IRNK)
      DIMENSION UU(IRNK,1),VV(IRNK,1)
      IF(IMAY.EQ.2.OR.IMAY.EQ.4)PRINT*,"THE LEFT SINGULAR"
      IF(IMAY.EQ.2.OR.IMAY.EQ.4)PRINT*,"VECTORS ARE "
      IF(IMAY.EQ.2.OR.IMAY.EQ.4)CALL PRINTM(N1,N2,UU)
      IF(IMAY.EQ.3.OR.IMAY.EQ.4)PRINT*,"THE RIGHT SINGULAR"
      IF(IMAY.EQ.3.OR.IMAY.EQ.4)PRINT*,"VECTORS ARE "
      IF(IMAY.EQ.3.OR.IMAY.EQ.4)CALL PRINTM(N2,N2,VV)
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE READS IN A GIVEN MATRIX
C*****************************************************************
      SUBROUTINE READM(M,N,RRMAT)
      DIMENSION RRMAT(10,10)
      PRINT*,"ENTER THE MATRIX BY ROWS"
100   DO 115 I=1,M,1
10    WRITE(6,15) I
15    FORMAT(1X,"ROW(",I2,")=")
      READ*,(RRMAT(I,J),J=1,N,1)
115   CONTINUE
110   RETURN
      END
```

```
C*****************************************************************
C     THIS SUBROUTINE ALLOWS USER TO CHANGE ERRONEOUS DATA
C*****************************************************************
      SUBROUTINE MISTAKE(M,N,RMAT)
      DIMENSION RMAT(10,10)
      MIST=2
      PRINT*,"DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) > "
      READ*,INDIC
      IF(INDIC.EQ.2)GO TO 2
3     PRINT*,"TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
      MIST=1
      PRINT*,"THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
      PRINT*,"MATRIX ELEMENT SEPARATED BY COMMAS > "
      READ*,MM,NN,CORREC
      RMAT(MM,NN)=CORREC
      PRINT*,"ANY MORE?(1=YES,2=NO) > "
      READ*,INDU
      IF(INDU.EQ.2)GO TO 2
      GO TO 3
2     IF(MIST.EQ.1)CALL PRINTM(M,N,RMAT)
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C*****************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H1,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SING VALUES
C*****************************************************************
      SUBROUTINE PRINTM1(M,N,DMAT)
      DIMENSION DMAT(30,1)
      DO 20I=1,M,1
      WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
```

```
10    FORMAT(1HC,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C***************************************************************
      OVERLAY(15,0)
C***************************************************************
C     THIS PROGRAM PLOTS THE FREQUENCY RESPONSE TO A USER
C     SPECIFIED ORDER BALANCED SYSTEM PROVIDED BY OPTION 2.
C
C
      PROGRAM FREAK
      LOGICAL SUP
      INTEGER NHALF,WINDIC
      REAL MAG,MAGG(102),IMAG
      COMPLEX EIG(10),EIGV(10,10),REIGV(10,10),RES(10),GG(100),
     +BB(10,10),BC(100),SUMB,SUMC
      DIMENSION ANGG(102),WK(20),ID1(17),ID2(17)
      DIMENSION W(102)
      COMMON/MIMO/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,1
     +),MC,NC
      COMMON/MIMO2/DBAK(10,10),SIGU2IN(10,10),Z(10,10),CBAK(10,10),TINV(
     +10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      EQUIVALENCE (BB(1,1),BC(1))
      DATA ID1(1)/20H  MAGNITUDE PLOT    /
      DATA ID1(3)/20H   VS. W(RAD/SEC)   /
      DATA ID1(5)/20H OF GIVEN ORDER     /
      DATA ID1(7)/20H TO INDIVIDUAL STATE/
      DATA ID1(9)/20H     W (RAD/SEC)    /
      DATA ID1(11)/20H   MAGNITUDE (DB)   /
      DATA ID2(1)/20H  PHASE PLOT        /
      DATA ID2(3)/20H   VS. W(RAD/SEC)   /
      DATA ID2(5)/20HOF TRANSFER FUNCTION/
      DATA ID2(7)/20H OF GIVEN ORDER     /
      DATA ID2(9)/20H  W (RAD/SEC)       /
      DATA ID2(11)/20H  PHASE ANGLE (DEG) /
      SUP=.FALSE.
      DO 3MM=1,102
      MAGG(MM)=0.0E0
      ANGG(MM)=0.0E0
```

```
3     CONTINUE
      PRINT*,"FREQUENCY RESPONSE FOR BALANCED SYSTEM"
      PRINT*," "
      PRINT*,"THE BALANCED A MATRIX MUST BE CREATED BY OPTION"
      PRINT*,"3 BEFORE THIS OPTION CAN BE EXECUTED!!!"
2     PRINT*,"THIS OPTION PLOTS THE FREQUENCY RESPONSE TO A"
      PRINT*,"GIVEN ORDER SYSTEM. ENTER DESIRED ORDER > "
      READ*,ISTT
      IF(ISTT.LE.0.OR.ISTT.GT.N)PRINT*,"ILLEGAL ORDER"
      IF(ISTT.LE.0.OR.ISTT.GT.N)GO TO 2
      PRINT*,"SYSTEM HAS ",NP," INPUTS AND ",NC," OUTPUTS."
      PRINT*,"SELECT DESIRED INPUT-OUTPUT COMBINATION > "
      READ*,L,K
      PRINT*,"PLEASE ENTER WMIN, -3 FOR .001,-2 FOR .01 ETC > "
      READ*,INT
      WMIN=10.**INT
      DO 97 IO=1,10
      DO 97 JO=1,10
      CREAL(IO,JO)=AREAL(IO,JO)
97    CONTINUE
      CALL EIGRF(CREAL,ISTT,10,1,EIG,EIGV,10,WK,IER)
      DO 5 II=1,N
      DO 5 JJ=1,N
      REIGV(II,JJ)=(0.0E0,0.0E0)
      REIGV(II,II)=(1.0,0.)
      BB(II,JJ)=EIGV(II,JJ)
5     CONTINUE
      CALL LEQT1C(BB,ISTT,10,REIGV,ISTT,10,0,WK,IER)
      DO 4 MM=1,102
      MAGG(MM)=0.0E0
      ANGG(MM)=0.0E0
      DO 4 NP=1,10
      GG(NP)=(0.0E0,0.0E0)
      BC(NP)=(0.0E0,0.0E0)
4     CONTINUE
      DO 10 J=1,ISTT
      SUM3=(0.0E0,0.0E0)
      SUM4=(0.0E0,0.0E0)
      DO 9 I=1,ISTT
      SUM3=SUM3+(REIGV(J,I)*BBAR(I,L))
      SUMC=SUMC+(CBAR(K,I)*EIGV(I,J))
96    CONTINUE
      RES(J)=SUMC/SUMD
      WINDIC=1
      WMIN1=WMIN
      W(WINDIC)=WMIN1
80    BC(WINDIC)=RES(J)*(CMPLX(1.,0.)/(CMPLX(0.,W(WINDIC))-EIG(J)))
      GG(WINDIC)=GG(WINDIC)+BC(WINDIC)
      IF(WINDIC.GE.100)GO TO 100
      W(WINDIC+1)=W(WINDIC)+WMIN1
      WINDIC=WINDIC+1
      GO TO 80
100   CONTINUE
82    DO 92KL=1,WINDIC
      CALL POLAR(MAG,ANG,GG(KL))
      MAG=20.*ALOG10(MAG)
      MAGG(KL)=MAG
      ANGG(KL)=ANG
92    CONTINUE
      PRINT *,L," - ",K," INPUT-OUTPUT COMBINATION TRANSFER FUNCTION"
      PRINT*,"FREQUENCY RESPONSE PLOT FOR ",ISTT," ORDER SYSTEM"
      PRINT*,"MAGNITUDE PLOT (DB)"
      CALL TITLES(ID1(13))
      PRINT*,"PHASE PLOT(DEG)"
      CALL TITLES(ID2(13))
      CALL PLOTS(DATA,NUM,MUM)
      CALL PLOT(0.,-6.,-3)
      CALL PLOT(0.0,0.93,-3)
      CALL HGRAPH1(W,MAGG,WINDIC,ID1,1,0,0)
      CALL HGRAPH1(W,ANGG,WINDIC,ID2,1,0,0)
      CALL PLOTE(MDUM)
      WHALF=WINDIC/2
      PRINT*,"DO YOU WISH TO SUPRESS THE LISTING(1=YES,2=NO) > "
      READ*,IOPT
      IF(IOPT.EQ.1)SUP=.TRUE.
      IF(SUP)GO TO 9999
      WRITE(6,30)
30    FORMAT(1X,T6,"W",T23,"MAG(DB)",T39,"ANGLE(DEG)")
      DO 97KI=1,WINDIC
      WRITE(6,36) W(KI),MAGG(KI),ANGG(KI)
```

```
96    FORMAT(1X,T7,G7.2,T20,E13.4,T35,E13.4)
93    CONTINUE
5599  CONTINUE
      END
C*****************************************************************
C     THIS SUBROUTINE FINDS THE MAG AND ANGLE OF A COMPLEX NUMBER
C*****************************************************************
      SUBROUTINE POLAR(MAG,ANG,RATELE)
      COMPLEX RATELE
      REAL MAG,IMAG
      RMAG=REAL(RATELE)
      IMAG=AIMAG(RATELE)
      MAG=SQRT(RMAG**2+IMAG**2)
      IF(RMAG)6,1,5
1     IF(IMAG)2,3,4
2     ANG=-90.
      RETURN
3     ANG=0.
      RETURN
4     ANG=-270.
      RETURN
5     ANG=ATAN(IMAG/RMAG)*(180./3.1415926535)
      RETURN
6     ANG=ATAN(IMAG/RMAG)*(180./3.1415926535)-180.
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE PLOTS BODE PLOTS
C*****************************************************************
      SUBROUTINE HGRAPH1(X,Y,N,ID,NO,NP,NS)
      DIMENSION X(1),Y(1),ID(1)   $IF(NO.EQ.2)GO TO 30
      IF(NO.LT.3)GO TO 10
      CALL SCALE(X,7.,N,1)  $ CALL SCALE(Y,5.,N,1)
10    CALL PLOT(8.5,0.,-3)   $ CALL PLOT(0.,11.,3)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-7.15,1.35,2)  $ CALL PLOT(-7.15,9.65,2)
      IF(ID(1).EQ.00)GO TO 25
      CALL PLOT(-7.05,9.55,3)  $ CALL PLOT(-7.05,7.55,2)
      DO 20 I=1,7,2
20    CALL SYMBOL(I*.1-6.9,7.85,.07,ID(I),90.,20)
```

```
      CALL PLOT(-7.85,7.55,3)  $ CALL PLOT(-6.85,7.55,2)
      CALL PLOT(-6.85,9.55,2)  $ CALL PLOT(-7.85,9.55,2)
      CALL PLOT(-7.15,3.65,3)
25    CALL PLOT(-1.35,9.65,2)  $ CALL PLOT(-1.35,1.35,2)
      CALL SYMBOL(-6.65,1.15,.1,ID(13),90.,58)
      CALL AXIS(-1.85,2.1,ID(3),-2.,7.,5.,X(N+1),X(N+2))
      CALL AXIS(-1.35,2.1,ID(11),27,5.,90.,Y(N+1),Y(N+2))
30    Y(N+2)=-Y(N+2)
      X(N+1)=X(N+1)-2.1*X(N+2)  $ Y(N+1)=Y(N+1)+1.35*Y(N+2)
      CALL LINE(X,Y,N,1,42,13)
      X(N+1)=X(N+1)+2.1*X(N+2)  $ Y(N+1)=Y(N+1)-1.35*Y(N+2)
      Y(N+2)=-Y(N+2)
      RETURN $ END
C*************************************************************
C     THIS SUBROUTINE ALLOWS VARIABLE TITLES
C*************************************************************
      SUBROUTINE TITLES(TITLE)
      INTEGER TITLE
      DIMENSION INC(80),TITLE(5)
1111  DO 10 I=1,5
10    TITLE(I)=0
      PRINT*,"> (--------ENTER TITLE  (58 CHARACTERS MAX)--------) <"
      PRINT*,"> "
      READ 1000,INC
1000  FORMAT(80R1)
      NSP=II=0
1     II=II+1
      IF(II.GT.80)GO TO 60
      NSP=NSP+1
      IF(INC(II).NE.55B)NSP=0
      IF(II.GT.58)GO TO 1
      JJ=(II-1)/10 +1
      TITLE(JJ)=TITLE(JJ)*64+INC(II)
      GO TO 1
60    IF(NSP.GE.22)GO TO 65
      PRINT*,"TITLE TRUNCATED TO 58 CHARACTERS"
      PRINT 1001,TITLE
      GO TO 1111
65    LETTER=II-NSP
      PRINT 1001,TITLE
```

```
1001  FORMAT(3X,5A10)                                                        019700
      RETURN                                                                 019710
      END                                                                    019720
C*************************************************************              019730
      OVERLAY(11,0)                                                          019740
C*************************************************************              019750
C     THIS PROGRAM DISPOSES ALL ACCUMULATED PLOTS TO THE AFIT PLOTTER        019760
C                                                                            019770
C                                                                            019780
      PROGRAM DISPOS                                                         019790
      COMMON/DSS/NDSP                                                        019800
      PRINT*,"THIS OPTION DISPOSES ACCUMULATED PLOTS TO THE AFIT"            019810
      PRINT*,"TERMINAL UNDER YOUR 2 LETTER USER ID."                         019820
      CALL DSP(2HPB)                                                         019830
      PRINT*,"PLOT(S) DISPOSED"                                              019840
      NDSP=NDSP+1                                                            019850
      END                                                                    019860
C*************************************************************              019870
      OVERLAY(12,0)                                                          019880
C*************************************************************              019890
C     THIS PROGRAM SAVES ALL IMPOTANT INFORMATION IN A LOCAL FILE            019900
C     MEMORE SO THAT THE USER MAY STOP EXECUTION OF THE PROGRAM TO           019910
C     ACCOMPLISH OTHER TASKS AND THEN RECOVER TO THE POINT WHEN HE           019920
C     LEFT THE PROGRAM.                                                      019930
C                                                                            019940
C                                                                            019950
      PROGRAM SAVE                                                           019960
      LOGICAL SUP                                                            019970
      DIMENSION COM1(10,10)                                                  019980
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                      019990
      COMMON/MIMO1/LOPT,NSTAT,MP                                             020000
      COMMON/MIMO2/TMAT(100),SOMAX(100),SOMIN(100),SOMAX(100),SOMIN(100)020010
     +,CONMOD(100),CONCO(100),HMAX(100),HMIN(100),HCOND(100)                 020020
      COMMON/MIMO3/KMOB(30,10),KMOD(10,30),RMCC(30,10),N1,N2,UU(30,30)       020030
     +,VV(30,30),HHKL(30,30),HHKLI(30,30)                                    020040
      COMMON/MIMO5/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10020050
     +),MC,NC                                                                020060
      COMMON/MIMO7/BBAR(10,10),SIGO2IN(10,10),Z(10,10),CBAR(10,10),TINV(020070
     +10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)                          020080
      COMMON/MIMO8/SIGM(10,10),SIGM2(10,10),HINF(10,10),UH(10,10),           020090
     $VMTRAN(10,10),SIGC2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      COMMON/PLOTS1/I,SIVS(30,1),SUP,IWAS,JWAS,IWAS1,JWAS1
      COMMON/SAVE1/NTEST
      IF(LOPT.EQ.11)GO TO 100
      REWIND 9
      NTEST=969
      WRITE(9,*) NTEST
      WRITE(9,*) LOPT,A,B,C,F,G,HNKL,AREAL,BREAL,CREAL,BBAR,CBAR
      WRITE(9,*) THAT,SCMAX,SCMIN,SOMAX,SOMIN,CONDC,CONDO,HMAX
      WRITE(9,*) HMIN,HCOND,RMOB,RMOD,RMCC,N1,N2,M,N,MB,NB,MC,NC
      WRITE(9,*) SIVS,MP,HINF
      REWIND 9
      PRINT*,"ALL DATA UPDATED IN LOCAL FILE MEMOKE"
      GO TO 999
100   REWIND 9
      READ(9,*) TEST
      IF(EOF(9))105,110
110   NTEST=TEST
      IF(NTEST.EQ.969)GO TO 111
105   PRINT*,"MEMOKE CONTAINS NO INFO---OPTION ABORTED"
      GO TO 999
111   CONTINUE
      READ(9,*) LOPT,A,B,C,F,G,HNKL,AREAL,BREAL,CREAL,BBAR,CBAR
      READ(9,*) THAT,SCMAX,SCMIN,SOMAX,SOMIN,CONDC,CONDO,HMAX
      READ(9,*) HMIN,HCOND,RMOB,RMOD,RMCC,N1,N2,M,N,MB,NB,MC,NC
      WRITE(9,*) SIVS,MP,HINF
      LOPT=1
999   CONTINUE
      END
C********************************************************************
      OVERLAY(13,0)
C********************************************************************
C     THIS PROGRAM STORES THE (A,B,C) AND (A*,B*,C*) SYSTEMS IN SUCH
C     A WAY THAT MIMO CAN DIRECTLY INTERFACE WITH TOTAL WITHOUT THE
C     USER HAVING TO RETYPE THE SYSTEMS IF HE DESIRES OPTIONS THAT
C     TOTAL OFFERS.
C
C
      PROGRAM MEMA
      DIMENSION AAA(100),BBB(100),CCC(100)
      DIMENSION CLNPOLY(307)                                            020500
      COMMON/AR/AA(100),MRA,NRA,BB(100),MRB,NRB,CC(100),MRC,NRC         020510
      COMMON/MIMO/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10020520
     *),MC,NC                                                           020530
      COMMON/MIMO2/BBAR(10,10),SIGOZIN(10,10),Z(10,10),CBAR(10,10),TINV(020540
     *10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)                     020550
      COMMON/SP/SMAT(10,10),MS,NS                                       020560
      COMMON/MASSTO/INMASS(47)                                          020570
      EQUIVALENCE(AA(1),CLNPOLY(1))                                     020580
      EQUIVALENCE(BB(1),CLNPOLY(103))                                   020590
      EQUIVALENCE(CC(1),CLNPOLY(205))                                   020600
      EQUIVALENCE(AAA(1),AREAL(1,1))                                    020610
      EQUIVALENCE(BBB(1),BREAL(1,1))                                    020620
      EQUIVALENCE(CCC(1),CREAL(1,1))                                    020630
      DO 10I=1,10                                                       020640
      DO 10J=1,10                                                       020650
      SMAT(I,J)=0.0E0                                                   020660
10    CONTINUE                                                          020670
      MS=NS=0                                                           020680
      DO 20I=1,307                                                      020690
      CLNPOLY(I)=0.0E0                                                  020700
20    CONTINUE                                                          020710
      CALL OPENMS(8,INMASS,47,0)                                        020720
      DO 11102I=1,25                                                    020730
      IF(INMASS(I+1).EQ.0)CALL WRITMS(8,CLNPOLY,307,I,-1)               020740
11102 CONTINUE                                                          020750
      DO 11103I=26,45                                                   020760
      IF(INMASS(I+1).EQ.0)CALL WRITMS(8,SMAT,102,I,-1)                  020770
11103 CONTINUE                                                          020780
      DO 1K=1,100                                                       020790
      AA(K)=AAA(K)                                                      020800
      BB(K)=BBB(K)                                                      020810
      CC(K)=CCC(K)                                                      020820
1     CONTINUE                                                          020830
      MRA=M                                                             020840
      NRA=N                                                             020850
      MRB=MB                                                            020860
      NRB=NB                                                            020870
      MRC=MC                                                            020880
      NRC=NC                                                            020890
      CALL WRITMS(8,A(1),132,33,-1,0)
      CALL WRITMS(8,B(1),102,40,-1,0)
      CALL WRITMS(8,C(1),132,41,-1,0)
      CALL WRITMS(8,AA(1),132,42,-1,0)
      CALL WRITMS(8,BB(1),132,43,-1,0)
      CALL WRITMS(8,CC(1),132,44,-1,0)
      CALL CLOSMS(8)
      PRINT*,"(A,B,C) AND (A*,B*,C*) HAVE BEEN STORED IN SUCH A WAY"
      PRINT ,"THAT TOTAL CAN DIRECTLY ACCESS THEM."
      PRINT*,"THROUGH USE OF THE COPY COMMAND UNDER TOTAL FULL FLEXI-"
      PRINT*,"BILITY IS EXERCISED. THE FOLLOWING LIST IS THE KEY"
      PRINT*," "
      PRINT*,"      MIMO                        STORED IN TOTAL"
      PRINT*,"      A ORIGINAL                   UMAT"
      PRINT*,"      B ORIGINAL                   VMAT"
      PRINT ,"      C ORIGINAL                   WMAT"
      PRINT*,"      A BALANCED                   XMAT"
      PRINT*,"      B BALANCED-OUTPUT            YMAT"
      PRINT*,"      C BALANCED-OUTPUT            ZMAT"
      PRINT*," "
      PRINT*," USING TOTAL'S OPTION 25 THE TRANSFER FUNCTION MAY BE "
      PRINT*,"OBTAINED. WHEN MIMO IS TERMINATED THE ABOVE INFOPMATION"
      PRINT*,"IS STORED IN MASS STORAGE FORMAT ON THE LIKEWISE TOTAL"
      PRINT*,"NAMED FILE-----MEMAUX---THIS FILE MUST EXIST PRIOR TO "
      PRINT*,"TRYING TO INTERFACE WITH TOTAL!!!!!!!!"
      PRINT ,"THE FOLLOWING PROCEDURE SHOULD BE UTILIZED."
      PRINT*,"FIRST COPY (UMAT,VMAT,WMAT) OR (XMAT,YMAT,ZMAT) INTO"
      PRINT ,"(AMAT,BMAT,CMAT).  THEN TYPE OPTION 19 IN TOTAL."
      PRINT*,"WHEN ASKED FOR THE NUMBER OF STATES--ENTER THE FULL"
      PRINT*,"NUMBER IF USING THE ORIGINAL SYSTEM.  ENTER THE DESIRED"
      PRINT ,"ORDER IF USING THE BALANCE SYSTEM. THEN FOR EACH"
      PRINT*,"ELEMENT, TYPE A *(ASTERIK) WHICH MEANS TOTAL WILL KEEP "
      PRINT ,"THAT LOCATION THE SAME. THEN ENTER NO FOR THE DMAT AND"
      PRINT ,"KMAT IF APPROPRIATE. THEN OPTION 25 CAN BE EXECUTED."
      END
C#####################################################################
      OVERLAY(14,0)
C#####################################################################
C     THIS PROGRAM USES A NOSLIB SUBROUTINE TO REQUEST A QUEUE
C     DEVICE FOR THE PLOT FILE AS IS NECESSARY FOR THE AFITSUB
```

```
C     NDSP TO DISPOSE THE PLOTS TO THE PLOTTER(OPTION 9)
C
C
      PROGRAM SET
      COMMON/DSS/NDSP
      IF(NDSP.EQ.1)CALL REQUEST(4HPLOT,2H Q)
      END
C*****************************************************************
      OVERLAY(15,7)
C*****************************************************************
C     THIS PROGRAM FINDS THE OUTPUT-NORMAL REP.
C
C
      PROGRAM OUTPAL
      DIMENSION APAR(10,10)
      COMMON/MIMO3/RMOB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,UU(30,30)
     +,VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO/WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10
     +),MC,NC
      COMMON/MIMO/BBAR(10,10),SIGO2IN(10,10),Z(10,10),CBAR(10,10),TINV(
     +10,10),AREAL(10,10),BREAL(10,10),CREAL(10,10)
      COMMON/MIMO/SIGH(10,10),SIGH2(10,10),HINF(10,10),UH(10,10),
     +VHTRAN(10,10),SIGC2(10,10),SIGO2(10,10),VOUTRAN(10,10)
      PRINT*,"THIS OPTION FINDS THE OUTPUT-NORMAL REP."
      PRINT*,"IT ALSO FINDS THE CORRESPONDING (F,G,C) SYSTEM"
      PRINT*,"AT THE USER'S OPTION-THE OBS. MAT. MAY BE FORMED."
      DO 10I=1,M
      DO 10J=1,N
      TINV(I,J)=SIGH2(I,J)
10    CONTINUE
      CALL LINV1F(TINV,N,10,SIGO2IN,1,WORAREA,IER)
      CALL MMULT(SIGH2,BREAL,BBAR,N,MB,NB)
      CALL MMULT(SIGH2,AREAL,WKAREA,N,N,N)
      CALL MMULT(WKAREA,SIGO2IN,ABAR,N,N,N)
      CALL MMULT(CREAL,SIGO2IN,CBAR,MC,NC,N)
      PRINT ,"THE ABAR MATRIX IS "
      CALL PRINTM(N,N,ABAR)
      PRINT*,"THE BBAR MATRIX IS "
      CALL PRINTM(MB,NB,BBAR)
      PRINT*,"THE CBAR MATRIX IS "
      CALL PRINTM(MC,NC,CBAR)
      PRINT*,"PLEASE ENTER THE SAMPLING TIME > "
      READ*,TSAMP
      CALL DKREET(F,G,N,MB,NB,WKAREA,ABAR,BBAR,TSAMP)
      PRINT*,"THE F MATRIX IS "
      CALL PRINTM(N,N,F)
      PRINT*,"THE G MATRIX IS "
      CALL PRINTM(MB,NB,G)
      PRINT*,"DO YOU WISH TO OBTAIN THE DISCRETE OBS. MAT(1=YES,2=NO)> "
      READ*,INDIC
      IF(INDIC.EQ.2)GO TO 9999
      CALL OBS(MC,NC,N,F,WINMED,WORAREA,FMOB,CBAR,N2)
      PRINT*,"THE OBS. MAT. IS "
      CALL PRINTM(N2,N,FMOB)
9999  CONTINUE
      END
C***************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C***************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20I=1,M,1
      WRITE (6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C***************************************************************
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C***************************************************************
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.000
      DO 1J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C*****************************************************************
      SUBROUTINE PRINTM1(M,N,EMAT)
      DIMENSION EMAT(30,10)
      DO 20I=1,M,1
      WRITE (6,10) I,(EMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE FINDS EXP(AMAT*T) AND ITS INTEGRAL
C     VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C*****************************************************************
      SUBROUTINE DISKRET(N,A,DEL,EA,EAINT,NT)
      DIMENSION EA(10,10),EAINT(10,10),A(10,10),Z(10,10),
     CHINMED(10,10),CBAR(10,10)
      DO 100 I=1,N
      DO 100 J=1,N
      EAINT(I,J)=0.00D
      EAINT(I,I)=DEL
      Z(I,J)=EAINT(I,J)
100   CONTINUE
      DO 400 K=2,NT
      DO 150 I=1,N
      DO 150 J=1,N
      CBAR(I,J)=A(I,J)*(DEL/FLOAT(K))
150   CONTINUE
      CALL MMULT(Z,CBAR,WINMED,N,N,N)
      DO 160 I=1,N
      DO 160 J=1,N
      Z(I,J)=WINMED(I,J)
      EAINT(I,J)=EAINT(I,J)+WINMED(I,J)
160   CONTINUE
400   CONTINUE
```

```
      CALL MMULT(A,EAINT,Z,4,N,N)
      DO 500 I=1,N
      DO 500 J=1,N
      EA(I,J)=0.0D0
      EA(I,I)=1.0D0
500   CONTINUE
      DO 600 I=1,N
      DO 600 J=1,N
      EA(I,J)=EA(I,J)+Z(I,J)
600   CONTINUE
      RETURN
      END
C*********************************************************************
C     THIS SUBROUTINE ACCEPTS EXP(AMAT*T) AND ITS INTEGRAL
C     AND FORMS THE DISCRETE F MATRIX AND G MATRIX
C*********************************************************************
      SUBROUTINE FKREFT(F,G,N,MB,NB,WKAREA,A,B,T)
      DIMENSION F(10,10),G(10,10),WKAREA(10,10),A(10,10)
      DIMENSION B(10,10)
      CALL DISKMEX(N,A,T,F,WKAREA,50)
      CALL MMULT(WKAREA,B,G,N,MB,NB)
      RETURN
      END
C*********************************************************************
C     THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C     MATRIX
C*********************************************************************
      SUBROUTINE CONBTY(F,G,MB,NB,N,PMDD,WORAKEA,WINMED,N1)
      DIMENSION F(10,10),G(10,10),PMDD(10,30),WINMED(10,10)
      DIMENSION WORAKEA(10,10)
      DO 100 I=1,MB
      DO 100 J=1,NB
      PMDD(I,J)=G(I,J)
100   CONTINUE
      N1=NB
      IF(N.EQ.1)GO TO 150
      CALL MMULT(F,G,WINMED,N,N,NB)
      DO 12 I=1,MB
      DO 12 J=1,NB
      K=J
      N1=N9
      KMOD(I,N9+J)=WINMED(I,J)
120   CONTINUE
      N1=N1+K
      IF(N.EQ.2)GO TO 150
      NM=N-1
      DO 140 L=2,NM,1
      CALL MMULT(F,WINMED,WORAREA,N,N,NB)
      DO 130 I=1,MC,1
      DO 130 J=1,NC,1
      KMOD(I,NL+J)=WORAREA(I,J)
      WINMED(I,J)=WORAREA(I,J)
      K=J
130   CONTINUE
      N1=N1+K
140   CONTINUE
150   RETURN
      END
C*******************************************************************
C     THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX
C*******************************************************************
      SUBROUTINE OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)
      DIMENSION WINMED(10,10),WORAREA(10,10),C(10,10)
      DIMENSION F(10,10),RMOB(30,10)
      DO 200 I=1,MC
      DO 200 J=1,NC
      RMOB(I,J)=C(I,J)
      N2=MC
200   CONTINUE
      IF(N.EQ.1)GO TO 250
      CALL MMULT(C,F,WINMED,MC,NC,N)
      DO 220 I=1,MC
      DO 220 J=1,NC
      K=I
      N2=MC
      RMOB(N2+I,J)=WINMED(I,J)
220   CONTINUE
      N2=N2+K
      IF(N.EQ.2)GO TO 250
      NM=N-1
      DO 240 L=2,NN,1
      CALL MMULT(WINMED,F,WORAREA,MC,NC,N)
      DO 230 I=1,MC
      DO 230 J=1,NC
      RMOB(N2+I,J)=WORAREA(I,J)
      WINMED(I,J)=WORAREA(I,J)
      K=I
230   CONTINUE
      N2=N2+K
240   CONTINUE
250   RETURN
      END
C#################################################################
      OVERLAY(15,0)
C#################################################################
C     THIS PROGRAM FINDS THE OUTPUT PREDICTIVE REP.
C
C
      PROGRAM OUTPRED
      DIMENSION P(10,10)
      DIMENSION ABAR(10,10),BBAR(10,10),CBAR(10,10)
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH
      COMMON/MIMO2/RMOB(30,10),RMOD(10,30),RMCC(30,10),N1,N2,UU(30,30)
     +,VV(30,30),HNKL(30,30),HNKLT(30,30)
      COMMON/MIMO1/WORAREA(10,10),WKAREA(10,10),WINMED(10,10)
      COMMON/MIMO/A(10,10),M,N,B(10,10),MB,ND,F(10,10),G(10,10),C(10,10
     +),MC,NC
      PRINT*,"THIS OPTION FINDS THE OUTPUT PREDICTIVE REP."
C
C
      PRINT*,"ENTER SAMPLING TIME > "
      READ*,TSAMP
      CALL DKRFFT(F,G,N,MB,NP,WKAREA,A,P,TSAMP)
      CALL OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)
      IF(N2.NE.N)PRINT*,"OPTION ABORTED--MUST BE SISO"
      IF(N2.NE.N)GO TO 9999
      DO 10 I=1,N2
      DO 10 J=1,N
      RMCC(I,J)=RMOB(I,J)
10    CONTINUE
```

```
      CALL SMINV(M2,N,RMCC,P,M2,0)
      CALL VMULFF(RMOD,F,N2,N,N,10,10,WINMED,10,IER)
      CALL VMULFF(WINMED,P,N2,N,N2,10,10,ABAR,10,IER)
      CALL VMULFF(RMUS,G,N2,N3,NB,10,10,BBAR,10,IER)
      CALL VMULFF(C,P,MC,NC,N2,10,10,CBAR,10,IER)
      PRINT*,"THE F' MATRIX IS "
      CALL PRINTM(N2,N2,ABAR)
      PRINT*,"THE G' MATRIX IS "
      CALL PRINTM(N2,N3,BBAR)
      PRINT*,"THE C' MATRIX IS "
      CALL PRINTM(MC,N2,CBAR)
9999  CONTINUE
      END
C*****************************************************************
C     THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C*****************************************************************
      SUBROUTINE PRINTM(M,N,DMAT)
      DIMENSION DMAT(10,10)
      DO 20I=1,M,1
      WRITE (6,10) I,(DMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C*****************************************************************
      SUBROUTINE MMULT(A,B,C,L,M,N)
      DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
      DO 2I=1,L
      DO 2K=1,N
      SUM=0.000
      DO 1J=1,M
1     SUM=SUM+A(I,J)*B(J,K)
2     D(I,K)=SUM
      DO 3I=1,L
      DO 3J=1,N
3     C(I,J)=D(I,J)
      RETURN
      END
```

```
C*****************************************************************
C     THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C*****************************************************************
      SUBROUTINE PRINTM1(M,N,EMAT)
      DIMENSION EMAT(30,30)
      DO 20 I=1,M,1
      WRITE (6,10) I,(EMAT(I,J),J=1,N,1)
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))
20    CONTINUE
      RETURN
      END
C*****************************************************************
C     THIS SUBROUTINE FINDS EXP(AMAT*T) AND ITS INTEGRAL
C     VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C*****************************************************************
      SUBROUTINE DISKRET(N,A,DEL,EA,EAINT,NT)
      DIMENSION EA(10,10),EAINT(10,10),A(10,10),Z(10,10),
     CWINMED(10,10),CBAR(10,10)
      DO 100 I=1,N
      DO 100 J=1,N
      EAINT(I,J)=0.000
      EAINT(I,I)=DEL
      Z(I,J)=EAINT(I,J)
100   CONTINUE
      DO 400 K=2,NT
      DO 150 I=1,N
      DO 150 J=1,N
      CBAR(I,J)=A(I,J)*(DEL/FLOAT(K))
150   CONTINUE
      CALL MMULT(Z,CBAR,WINMED,N,N,N)
      DO 160 I=1,N
      DO 160 J=1,N
      Z(I,J)=WINMED(I,J)
      EAINT(I,J)=EAINT(I,J)+WINMED(I,J)
160   CONTINUE
400   CONTINUE
      CALL MMULT(A,EAINT,Z,N,N,N)
      DO 500 I=1,N
      DO 500 J=1,N
      EA(I,J)=0.000
      EA(I,I)=1.0D0
500   CONTINUE
      DO 600I=1,N
      DO 600 J=1,N
      EA(I,J)=EA(I,J)+Z(I,J)
600   CONTINUE
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE ACCEPTS EXP(AMAT*T) AND ITS INTEGRAL
C     AND FORMS THE DISCRETE F MATRIX AND G MATRIX
C*************************************************************
      SUBROUTINE DKRLET(F,G,N,MB,NB,WKAREA,A,B,T)
      DIMENSION F(10,10),G(10,10),WKAREA(10,10),A(10,10)
      DIMENSION B(10,10)
      CALL DISKRET(N,A,T,F,WKAREA,50)
      CALL MMULT(WKAREA,B,G,N,MB,NB)
      RETURN
      END
C*************************************************************
C     THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C     MATRIX
C*************************************************************
      SUBROUTINE CONDTY(F,G,MB,NB,N,RMOD,WORKAREA,WINMED,N1)
      DIMENSION F(10,10),G(10,10),RMOD(10,30),WINMED(10,10)
      DIMENSION WORKAREA(10,10)
      DO 100 I=1,MB
      DO 100 J=1,NB
      RMOD(I,J)=G(I,J)
100   CONTINUE
      N1=NB
      IF(N.EQ.1)GO TO 150
      CALL MMULT(F,G,WINMED,N,N,NB)
      DO 120 I=1,MB
      DO 120 J=1,NB
      K=J
      N1=NB
      RMOD(I,NB+J)=WINMED(I,J)
120   CONTINUE
      N1=N1+K
      IF(N.EQ.2)GO TO 150                                               024900
      NN=N-1                                                            024910
      DO 140 L=2,NN,1                                                   024920
      CALL MMULT(F,WINMED,WORAREA,N,N,NB)                               024930
      DO 130 I=1,N,1                                                    024940
      DO 130 J=1,NB,1                                                   024950
      RMOB(I,N1+J)=WORAREA(I,J)                                         024960
      WINMED(I,J)=WORAREA(I,J)                                          024970
      K=J                                                               024980
130   CONTINUE                                                          024990
      N1=N1+K                                                           025000
140   CONTINUE                                                          025010
150   RETURN                                                            025020
      END                                                               025030
C*********************************************************************** 025040
C     THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX           025050
C*********************************************************************** 025060
      SUBROUTINE OBS(MC,NC,N,F,WINMED,WORAREA,RMOB,C,N2)                025070
      DIMENSION WINMED(10,10),WORAREA(10,10),C(10,10)                   025080
      DIMENSION F(10,10),RMOB(30,10)                                    025090
      DO 200 I=1,MC                                                     025100
      DO 200 J=1,NC                                                     025110
      RMOB(I,J)=C(I,J)                                                  025120
      N2=MC                                                             025130
200   CONTINUE                                                          025140
      IF(N.EQ.1)GO TO 250                                               025150
      CALL MMULT(C,F,WINMED,MC,NC,N)                                    025160
      DO 220 I=1,MC                                                     025170
      DO 220 J=1,NC                                                     025180
      K=I                                                               025190
      N2=MC                                                             025200
      RMOB(N2+I,J)=WINMED(I,J)                                          025210
220   CONTINUE                                                          025220
      N2=N2+K                                                           025230
      IF(N.EQ.2)GO TO 250                                               025240
      NN=N-1                                                            025250
      DO 240 L=3,NN,1                                                   025260
      CALL MMULT(WINMED,F,WORAREA,MC,NC,N)                              025270
      DO 230 I=1,MC                                                     025280
      DO 230 J=1,NC                                                     025290
      RMOD(N2+I,J)=WORAREA(I,J)
      WIMMED(I,J)=WORAREA(I,J)
      K=I
230   CONTINUE
      N2=N2+K
240   CONTINUE
250   RETURN
      END
C*************************************************
      OVERLAY(17,0)
C*************************************************
      PROGRAM GRAMM
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH
      COMMON/MEMO1/A(10,10),M,N,B(10,10),MB,NB,F(10,10),G(10,10),C(10,10
     +),MC,NC
      COMMON/MEMO11/ATRAN(10,10),BTRAN(10,10),CTRAN(10,10),WSQUAR(10,10)
     +,WSQUAR2(10,10),BPROD(10,10),CPROD(10,10)
      DO 6I=1,N
      DO 6J=1,N
      ATRAN(J,I)=A(I,J)
6     CONTINUE
      DO 7I=1,MB
      DO 7J=1,NB
      BTRAN(J,I)=B(I,J)
7     CONTINUE
      DO 8I=1,MC
      DO 8J=1,NC
      CTRAN(J,I)=C(I,J)
8     CONTINUE
      CALL VMULFF(B,BTRAN,MB,NB,MB,10,10,BPROD,10,IER)
      CALL VMULFF(CTRAN,C,NC,MC,NC,10,10,CPROD,10,IER)
      CALL MLINEQ(N,ATRAN,BPROD,WSQUAR,.01)
      CALL MLINEQ(NC,A,CPROD,WSQUAR2,.01)
      PRINT*," "
      PRINT*,"THE CONTROLLABILITY GRAMMIAN IS "
      CALL PRINTM(N,N,WSQUAR)
      PRINT*," "
      PRINT*,"THE OBSERVABILITY GRAMMIAN IS "
      CALL PRINTM(N,N,WSQUAR2)
      END
```

```
C*****************************************************         025700
C     THIS PROGRAM PRINTS OUT A GIVEN MATRIX                   025710
C*****************************************************         025720
      SUBROUTINE PRINTM(M,N,DMAT)                              025730
      DIMENSION DMAT(10,10)                                    025740
      DO 20 I=1,M,1                                            025750
      WRITE (6,10) I,(DMAT(I,J),J=1,N,1)                       025760
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))                        025770
20    CONTINUE                                                 025780
      RETURN                                                   025790
      END                                                      025800
```

VITA

James Robert McClendon was born on 12 October 1955 in Aruba, Netherlands Antilles. He was graduated from Spartanburg High School in 1973 and attended Clemson University from which he received a Bachelor of Science in Electrical and Computer Engineering with honors in May 1978. Upon graduation, he was designated a distinguished graduate and received a commission in the USAF through the AFROTC program. He entered active duty in June 1978 and was assigned to the School of Engineering, Air Force Institute of Technology.

Permanent Address: 155 Marlin Drive
Spartanburg, South Carolina
29302

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/79-22	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODEL ORDER REDUCTION USING THE BALANCED REPRESENTATION: THEORY, APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James R. McClendon 2nd Lt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1979 13. NUMBER OF PAGES 211
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release;
IAW AFR 190-17

Joseph P. Hipps, Major, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SINGULAR VALUE ANALYSIS
MODEL ORDER REDUCTION
SYSTEM MODELLING
INTERACTIVE COMPUTER PROGRAM
GRAMMIANS
MATHEMATICAL MODELLING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report investigates a model order reduction technique developed by Dr Bruce Moore of the University of Toronto, as applied to the B-52E flutter control problem currently under study by the Air Force Flight Dynamics Laboratory. The algorithm, which is based upon singular value analysis, is applied to the full twenty-fourth order model yielding an internally balanced representation which is balanced with respect to controllability and observability properties. The → over

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

system is reduced in order, and comparisons are made between the Moore algorithm model and that obtained via a method used by the Air Force Flight Dynamics Laboratory.

In addition to this investigation, an interactive program is presented which contains the model order reduction algorithm. Other capabilities include: estimation of the condition number with respect to inversion, singular value and condition number plotting vs. sample time for discrete time controllability, observability, and Hankel matrices, frequency response generation, and various special coordinate system transformations.